Unsupervised Category-Specific Partial Point Set Registration via Joint Shape Completion and Registration

Xiang Li, Lingjing Wang, and Yi Fang†

Abstract—We propose a self-supervised method for partial point set registration. Although recently proposed learning-based methods demonstrate impressive registration performance on full shape observations, these methods often suffer from performance degradation when dealing with partial shapes. To bridge the performance gap between partial and full point set registration, we propose to incorporate a shape completion network to benefit the registration process. To achieve this, we introduce a learnable latent code for each pair of shapes, which can be regarded as the geometric encoding of the target shape. By doing so, our model does not require an explicit feature embedding network to learn the feature encodings. More importantly, both our shape completion and point set registration networks take the shared latent codes as input, which are optimized simultaneously with the parameters of two decoder networks in the training process. Therefore, the point set registration process can benefit from the joint optimization process of latent codes, which are enforced to represent the information of full shapes instead of partial ones. In the inference stage, we fix the network parameters and optimize the latent codes to obtain the optimal shape completion and registration results. Our proposed method is purely unsupervised and does not require ground truth supervision. Experiments on the ModelNet40 dataset demonstrate the effectiveness of our model for partial point set registration.

Index Terms—Point Set Registration, Partial Registration, Unsupervised learning, Shape Completion.

1 Introduction

Point set registration is generally defined as the estimation of spatial transformation, either rigid or non-rigid transformation, to align one point set to another. It is a key component in various applications such as robotics, shape correspondence, and large-scale 3D reconstruction [1]. In recent decades, the point set registration problem has been actively pursued by the computer vision community, where a significant number of conventional and recent approaches were proposed [1], [2], [3], [4], [5].

For a pair of point sets, conventional non-learning-based methods mostly solve the registration problem by iteratively optimizing a predefined alignment loss between the transformed source point sets and their corresponding target sets. The best-known algorithm ICP [6] performs point set registration by iteratively estimating point correspondences and solves a rigid transformation via least-squares optimization. ICP variants [3], [7], [8], [9] try to enhance the performance from the perspective of input noise, partiality, sparsity, and efficiency. Probabilistic models [10], [11] have also been developed to handle uncertainty and partiality.

Owing to the prevalence of deep learning in various vision tasks in recent years, researchers have shifted their focus from "case-driven" iterative-based methods toward "data-driven" learning-based methods to perform point set registration. Recently proposed methods, such as PointNetLK [12] and DCP [1], show that deep neural networks with strong nonlinear modeling capacities exhibit superior performance compared with classical non-learning-based methods. During training, deep learning-based methods automatically learn to extract point features and their correspondences, and then solve the rigid or non-rigid transformations in an end-to-end manner. Most importantly, these methods exhibit good generalization abilities when transferred to unseen pairs, even when trained on different datasets. Learning-based methods significantly enhance registration efficiency and render it easier to perform registration at scale. Despite the significant success achieved, the performances of these methods are degraded when input point sets with missing components are involved.

More recently, PR-Net [13] proposed the first learning-based method for partial point set registration. PR-Net first detected keypoints from partial observations and then learned a mapping from the key points of the source shape to that of the target shape as the desired transformation. This method needs to refine the alignment in an iterative process, which makes the method time-consuming. Moreover, the registration performance is heavily dependent on the performance of keypoint detection.

In this study, we approached the partial point set registration task from a different perspective. Our method is motivated by two general findings: 1) The problem of point set registration on full point sets is a well-investigated problem, where satisfying and significantly better performances are achieved compared with the problem of partial point set registration. For example, the recently proposed DCP [1] method achieves a mean square error of 45.01° and 1.31° for rotation angle prediction on partial and full point sets from ModelNet40 dataset, respectively [5]; 2) recent progress in the deep learning field has seen numerous learning-based methods for shape completion from partial observations. These findings naturally motivated us to address the partial point set registration problem by first recovering full shapes from partial inputs. Therefore, in this paper, we propose to achieve partial point set registration by jointly optimizing two tasks: 1) To recover

Xiang Li, Lingjing Wang, and Yi Fang are with NYU Multimedia and Visual Computing Lab, NYU Tandon and Abu Dhabi, also with the Department of Electrical and Computer Engineering, NYU Tandon, USA, and with the Department of Electrical and Computer Engineering, NYU Abu Dhabi, UAE. Email:{xl1845,lingjing.wang,yfang}@nyu.edu.
† *Corresponding author: Yi Fang (yfang@nyu.edu).*

full 3D shapes from partial point sets, and 2) to learn a rigid transformation from the source point set to the target point set. Our model assumes that the source and target point sets are sampled from the same underlying surface, and that the target point set is transformed from the source point set under a rigid geometric transformation.

To integrate these two tasks for joint optimization, we introduce a learnable latent code for each pair of shapes that captures the geometric essence of the input shapes. Both our shape completion and point set registration networks use the shared latent code as input, which is optimized simultaneously with network parameters. Our shape completion network can thus enforce the optimized latent codes to represent the information of full shapes instead of partial inputs. Therefore, our point set registration network can benefit from the shape completion network during the joint optimization of the shared latent code. Considering that the shape completion network used in our method requires per-category training, the proposed method, in its current form, primarily focuses on modeled data from seen categories and is currently not evaluated on scans that are acquired from partial views of 3D scenes/models.

Our contributions are as below:

- We propose a novel unsupervised paradigm that jointly learns 3D shape completion and partial registration to integrate the completion task into the partial registration process to enhance performance.
- We introduce a novel learnable latent code to eliminate the explicit design of a specific 3D feature encoder network for irregular point clouds. The shared latent code enables the joint optimization of two coupling tasks: shape completion and registration.
- Experiments on the ModelNet40 benchmark dataset demonstrate the effectiveness of our model for partial point set registration.

2 Related Work

2.1 Point Set Registration

Point set registration is generally defined as finding the geometric transformations to align source point sets with target ones. Classical iterative optimization methods typically search for a set of optimal parameters of a transformation during the process of minimization of a predefined alignment loss. The Iterative Closest Point (ICP) algorithm [2] is a classical solution that has achieved significant success for rigid point set registration. The ICP algorithm leverages a set of corresponding points to define the transformation and iteratively refines the transformation by minimizing an error metric. In the ICP algorithm, the initial estimation of the desired rigid transformation can significantly affect the final performance. Yang et al. [3] proposed Go-ICP to solve the local initialization problem of ICP using a branch-and-bound (BnB) searching scheme over an entire 3D motion space. Gelfand et al. [14] developed a robust point set registration method that can be used without any assumptions regarding the initial positions. Krishnan et al. [15] proposed a novel algorithm for registering multiple 3D point sets within a common reference space using a manifold optimization approach. Mitra et al. [16] formulated the problem of aligning two point sets as a minimization of the distance between the underlying surfaces. Litany et al. [17] developed a method for simultaneously registering and segmenting multiple shapes based on a reference shape. In other methods such as TPS-RSM [18] and CPD [19], solutions were proposed to solve the non-rigid point set registration problem. More recently, Halimi et al. [20] claimed that the problem of partial matching and shape completion can be addressed in a holistic manner. One common disadvantage of these transitional methods is their registration efficiency. All of these methods register every single pair of source and target point sets as an independent optimization process, which prevents knowledge transfer from the registration of one pair to another.

Recently, deep learning methods have achieved significant success in many computer vision applications, including image classification [21], [22], semantic segmentation [23], [24], object detection [25], [26], [27], image registration [28], [29], and point signature learning [30], [31], [32], [33], [34]. Researchers have shifted their attention from classical methods to learning-based methods to address the point set registration problem [1], [12], [35], [36]. Aoki et al. [12] proposed PointNetLK to combine the Lucas & Kanade algorithm with the PointNet feature exaction module into a single trainable recurrent deep neural network. Liu et al. proposed FlowNet3D [36], which predicts the flow field to move the source point set towards the target one. The DCP [1] leverages a DGCNN structure for point feature learning, followed by an attention-based feature-matching module to generate point correspondences. RPM-Net [37] introduced a robust learning-based method for rigid point set registration, where a differentiable Sinkhorn layer and an annealing technique were applied to obtain soft assignments of point correspondences. Wang et al. [5] proposed PR-Net to learn partial registration. The PR-Net first leverages a keypoint detector to determine the corresponding points of the source and target point sets. Subsequently, based on the matched keypoints, the desired transformation can be further predicted. Unlike the DCP and PR-Net that use an explicit point feature encoding network to learn per-point features, in contrast, we represent the geometric features of 3D shapes using an optimizable latent code that enables joint shape completion and registration.

2.2 Shape Completion

3D shape completion is generally defined as the recovery of missing components from the original partial observations. It is a long-standing problem in the computer vision and graphics fields that has been investigated extensively, where both conventional non-learning-based and recent learning-based methods have been used. Conventional shape completion methods fill unseen components by assuming a local surface or volumetric smoothness. For example, one can fill holes with surface primitives, such as planes or quadrics, or cast the problem as an energy minimization process, e.g., Laplacian smoothing [38], [39]. Additionally, researchers have leveraged observed structures and regularities in 3D shapes, such as symmetries, to complete shapes [40], [41]. Another typically used strategy is to search for the most similar template shape from a database and align it with partial observations to achieve shape completion [42], [43]. These methods works well for objects with consistent topological structures, such as human faces [44], [45] and bodies [46], [47]. Nevertheless, these methods address every single shape independently; hence, they cannot transfer the local structural pattern or prior knowledge learned from one shape to another.

By contrast, learning-based methods leverage deep neural networks to learn shape completion patterns through network

training. These methods mostly adopt an encoder-decoder architecture to extract global feature representations from depth maps [48], RGB images [49], [50], discrete SDF voxels [51], or point clouds [52], and subsequently produce a reconstructed shape using the learned priors. Early studies [53], [54], [55] mostly use deep neural networks to predict discrete 3D shapes. For example, Yuan et al. [56] proposed a point cloud completion network that directly takes raw point clouds as inputs and produces dense complete point clouds using a coarse-to-fine decoder network. Similarly, Wen et al. [57] developed a point cloud completion method using a skip attention network with hierarchical folding. Meanwhile, other researchers [53], [54], [58], [59] have investigated generative models to generate 3D shapes. In [54], the authors developed a mesh variational auto-encoder (mesh VAE) to generate 3D meshes from a probabilistic latent space. Litany et al. [60] further enhanced the method by replacing the fully connected layers with graph convolutional layers. In more recent studies [55], [61], deep neural networks were developed to learn the implicit functions of 3D shapes in a continuous space. In [55], the authors proposed to use the signed distance function to represent 3D shapes. A generative model was trained to generate a continuous field for 3D shape representation, which was characterized by the zero iso-surface decision boundaries of discretized SDF samples. A similar idea was proposed in Occupancy Network [61] that uses the binary voxel occupancy to implicitly represent a 3D surface as a continuous decision boundary of discretized voxel occupancy.

2.3 Latent Space Optimization

Instead of using an explicit encoder network for feature learning, optimal latent representations can be searched by training a decoder-only network. Tan et al. [62] was the first to propose the abovementioned idea, which involves simultaneously optimizing the latent code and decoder network parameters through back-propagation. During inference, the decoder parameters are fixed as a prior, and a latent code is randomly initialized from a Gaussian distribution and optimized to reconstruct new observations. Similar approaches have been investigated recently [63], [64], [65], for applications such as noise reduction, missing measurement completions, fault detection, meta-learning, and shape correspondence. For example, in [65], the authors proposed to optimize the latent code extracted from an encoder network to minimize the chamfer distance between deformed shapes and target shapes, thereby improving the performance for shape correspondence. In [55], the authors proposed to search for optimal latent representations for shape completion. Herein, we refer to this type of encoder-free network as auto-decoders. In this study, both our point set registration and shape completion networks were developed based on auto-decoders.

3 METHOD

In this section, we introduce the proposed method for partial point set registration via **J**oint shape **C**ompletion and **R**egistration using a deep neural **Net**work, named JCRNet. Our method involves two networks: a shape completion network that recovers full shapes from corresponding partial observations, and a point set registration network that learns the rigid transformations from the source shapes to the target shapes. First, we define the partial registration problem in Section 3.1. An overview of the proposed method is presented in Section 3.2. We introduce our shape registration network and shape completion network in Section 3.4 and section 3.5 respectively. Finally, the training process is described in Section 3.6.

3.1 Problem Statement

For a source point set $\mathcal{X} = \{x_1, x_2, ..., x_{N_1}\} \subset \mathbb{R}^3$ and a target point set $\mathcal{Y} = \{y_1, y_2, ..., y_{N_2}\} \subset \mathbb{R}^3$, our point set registration model attempts to estimate the rigid transformation from source point set $\mathcal{X}$ to target point set $\mathcal{Y}$. In our partial registration setting, we primarily address the scenario where the source and target point sets are sampled from the same underlying surface $\mathcal{S} = \{s_1, s_2, ..., s_{N_f}\} \subset \mathbb{R}^3$, and we assume the target shape $\mathcal{Y}$ is transformed from source shape $\mathcal{X}$ by an unknown rigid transformation. Without loss of generality, the rigid transformation here is represented by a homogeneous transformation matrix, $\mathcal{T} \in SE(3)$, which is composed of a rotation matrix $R \in SO(3)$ and a translation vector $t \in \mathbb{R}^3$, i.e., $\mathcal{T} = [R|t]$. It it noteworthy that the source and target shapes are overlapped but do not necessarily exhibit one-to-one correspondences owing to different partial samplings. In the following sections, we assume $N_1 = N_2 = N$ for the ease of notation. It is noteworthy that our method can be used for $N_1 \neq N_2$ because none of the modules of our method require $\mathcal{X}$ and $\mathcal{Y}$ to have the same length or a bijective matching.

3.2 Method Overview

In this study, we designed two separate deep neural networks to learn shape completion and point set registration. To integrate these two tasks for joint optimization, we introduced a learnable latent code for each target shape, which can be regarded as a global encoding of the shape. Our method benefits from this design in two aspects: 1) First, we do not need to explicitly define a feature encoding network to learn the shape representation. Previous methods primarily rely on hand-crafted features or learning deep feature embeddings using deep neural networks. However, neither hand-crafted features nor a learning-based feature encoding network can guarantee robustness towards partial observations. 2) Both our shape completion and registration networks use the shared latent code representation as input, thereby enabling communication between these two tasks and hence improving registration performance. Therefore, both our shape registration and shape completion networks benefit from the design of the auto-decoder architecture. During training, we optimized all latent codes as well as the network parameters. In the inference stage, we fixed the network parameter and optimized the latent codes to obtain the optimal shape completion and registration results.

Fig. 1 provides an overview of the proposed method for partial point set registration. Our method contains two parts, both of which use an auto-decoder architecture. In the first part (upper section of Fig. 1), our method performs shape completion for the target shape $\mathcal{Y}$. It takes the latent code z as the input, and a decoder network g_β is designed to recover a full shape with the partial observation as supervision. In the second part (lower section of Fig. 1), our model learns the geometric transformation from the source point set $\mathcal{X}$ to the target point set $\mathcal{Y}$. Specifically, we stacked the coordinates of each point $x_i \in \mathcal{X}$ with latent vector z as the input, i.e., $[z, x_i]$, and fed them to another decoder network f_θ to predict a rigid transformation. By applying the predicted geometric transformation to the source shape, we then formulated an unsupervised alignment loss between the transformed source point set $\mathcal{X}_\mathcal{T}$ and the target point set $\mathcal{Y}$. The shared latent code

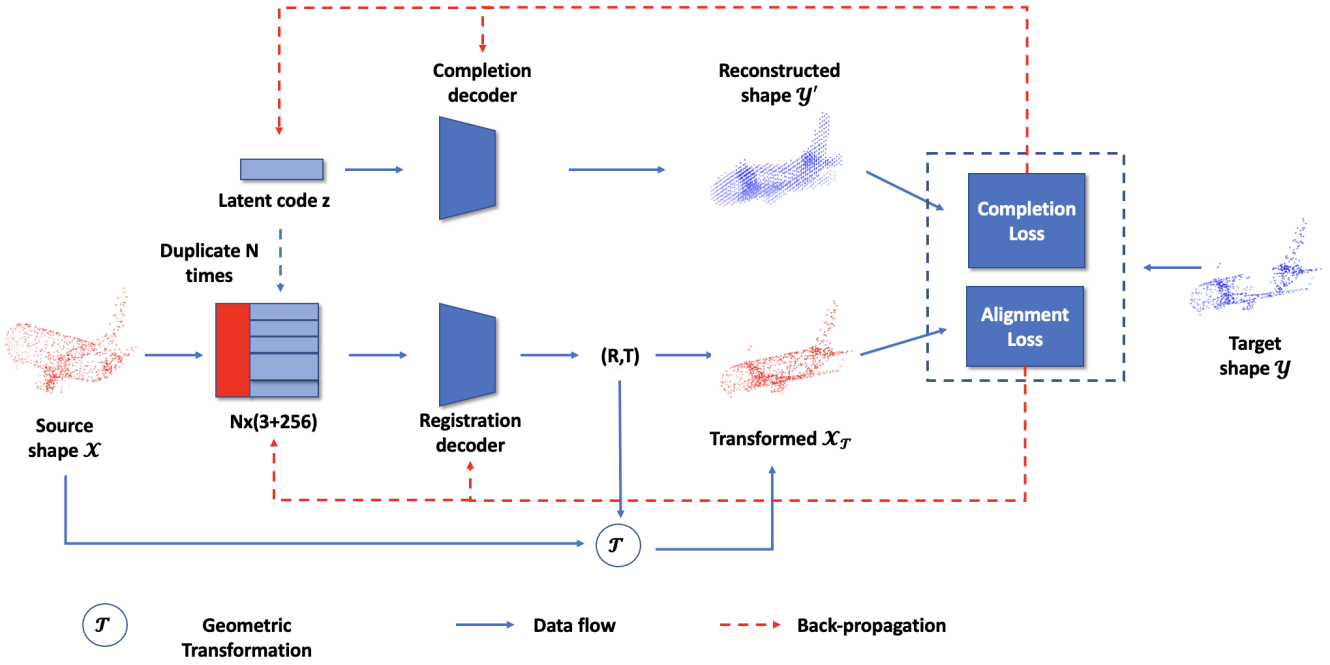

Fig. 1. Method overview. Our method contains two parts: a shape completion network that recovers full shapes from corresponding partial observations (upper section), and a point set registration network that performs point set registration (lower section). Both parts use an auto-decoder network that takes as input a shared latent code of target shape which is randomly initialized from a Gaussian distribution. More specifically, given a pair of input point sets, source point set $\mathcal{X} \in \mathbb{R}^{N\times 3}$ and target point set $\mathcal{Y} \in \mathbb{R}^{N\times 3}$, our model first introduces a learnable latent code z, which can be regarded as the global representation of the target point set. The shape completion network takes the latent code z as input and yields the recovered shape $\mathcal{Y}'$ under the supervision of $\mathcal{Y}$. The point set registration network takes the source point sets $\mathcal{X}$ and the latent code z as input and predicts the rigid transformation $\mathcal{T}$ from source point set $\mathcal{X}$ to target point set $\mathcal{Y}$. More specifically, we stack the coordinates of each point on shape $\mathcal{X}$ with the latent code z to formulate the inputs and use a fully connected network to regress the rigid transformation parameters. Our model achieves simultaneous shape completion and registration from partial observations.

z was initialized randomly from a Gaussian distribution. During training, all latent codes were optimized simultaneously with the two decoder networks. In the inference stage, we fixed the network parameters and optimized the latent code for each pair of input shapes to produce the optimal geometric transformation.

3.3 Global Feature Representation

To characterize the global feature of a given point set, a feature encoder network (i.e., PointNet [30]) is typically defined in previous methods to achieve high-level feature extraction. However, the design of an appropriate feature encoder for unstructured point clouds is an open research problem due to the irregular structures of raw point clouds. To avoid the hand-crafted network design, we herein introduced a learnable latent code for each target point set to characterize the essence of its global feature. As shown in Fig. 1, the latent code was randomly initialized from a zero-mean multivariate-Gaussian distribution with a spherical covariance $\sigma^2 I$; subsequently, it was fed into two different decoder networks for the joint tasks of shape completion and registration.

Based on the design of latent code representation, both our shape completion network and registration networks take as input the same global feature representation, which is optimized simultaneously with the network parameters during training. Hence, each latent code is enforced to characterize the global feature of a full shape instead of a partial one. In the inference stage, we fixed the network parameters and optimized only the latent code for each pair of input shapes.

It is plausible to use an explicit encoder network for shape representation learning. The auto-decoder architecture brings us three unique advantages for the shape completion: 1) It avoids the design of a specific 3D feature encoder network for irregular non-grid point clouds; 2) it enables fewer parameters to be trained compared with an encoder-decoder structure; 3) it enables further fine-tuning on the test data based on an unsupervised loss (e.g., reconstruction loss) to achieve a higher generalization ability of the model. By contrast, conventional neural networks do not exhibit flexibility in the test phase.

3.4 Point Set Registration

Our point set registration network f_θ takes the source point set $\mathcal{X}$ and the latent vector z of the target point set $\mathcal{Y}$ as inputs, and outputs the rigid transformation from $\mathcal{X}$ to $\mathcal{Y}$. The architecture of our point set registration network is illustrated in Fig. 2. For each point x_i on shape $\mathcal{X}$, we stacked its coordinates with the associated latent vector to formulate the input vector $[z, x_i]$. Subsequently, the inputs $\{[z, x_i]\}_{x_i\in\mathcal{X}}$ are then fed into the transformation decoder network f_θ to predict the desired transformation.

In this study, our transformation decoder network first uses a multi-layer perceptron (MLP) network to map the inputs to high-dimensional feature embeddings, followed by a max-pooling layer to obtain the global transformation representation. Several fully connected layers are further used to map the global transformation representation to the desired geometric transformation, i.e., three

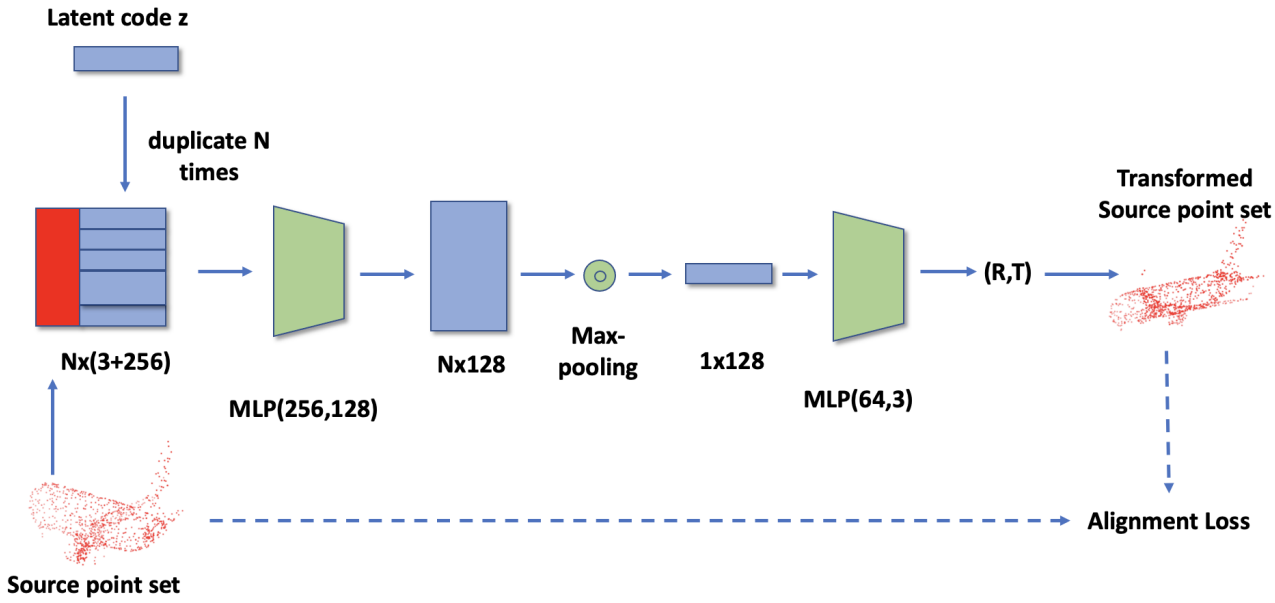

Fig. 2. Architecture of our point set registration network. Given a pair of source and target point sets with dimension of $N \times 3$, and the associated latent code with a dimension of 256, our model predicts the desired geometric transformation (R,T).

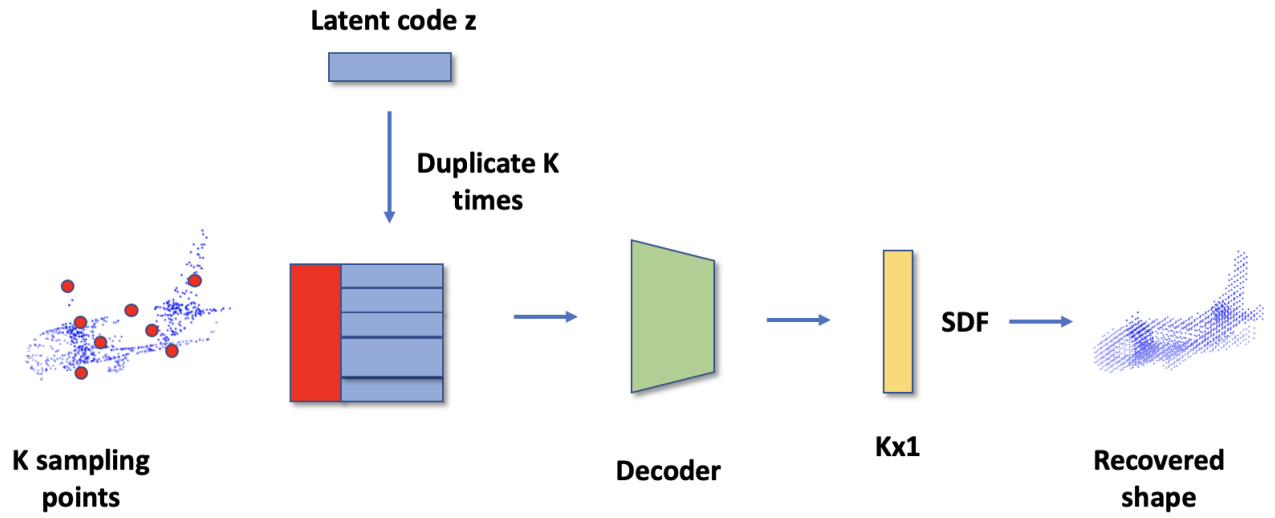

Fig. 3. Architecture of our shape completion network. Given a partial target point set and its associated latent code, our network produces K SDF values for K sampled spatial locations. A complete shape can be recovered by identifying the zero iso-surface decision boundary of discretized SDF samples.

rotation angles $r = (rx, ry, rz)$ and a translation vector $t = (ta, tb, tc)$.

$$[r, t] = f_\theta(\{[z, x_i]\}_{x_i \in \mathcal{X}}) \tag{1}$$

The rotation matrix R can then be formulated using the ration angles as follows:

$$R = Rz \odot Ry \odot Rx \tag{2}$$

where $\odot$ denotes matrix multiplication, Rx, Ry and Rz denote the rotation matrices along x-, y-, and z-axes, respectively, which are calculated from r.

The transformed source point set can be generated as follows:

$$\mathcal{X}_\mathcal{T} = R\mathcal{X} + t \tag{3}$$

Generally, the chamfer distance can be used to measure the alignment between two point sets. In this study, considering that the target point set is a partial observation, we used the clipped chamfer distance to measure the alignment loss between transformed source shape $\mathcal{X}_\mathcal{T}$ and partial target shape $\mathcal{Y}$, i.e., we penalized the distance from each point on partial target shape $\mathcal{Y}$ to the nearest point on transformed source shape $\mathcal{X}_\mathcal{T}$, and the distance from each point on transformed source shape $\mathcal{X}_\mathcal{T}$ to its nearest point on target shape $\mathcal{Y}$. The registration loss is calculated as follows:

$$\mathcal{L}_{reg}(\mathcal{X}_\mathcal{T}, \mathcal{Y}) = \sum_{y \in \mathcal{Y}} \min(\sigma_t, \min_{x \in \mathcal{X}_\mathcal{T}} ||x - y||_2^2) + \sum_{x \in \mathcal{X}_\mathcal{T}} \min(\sigma_t, \min_{y \in \mathcal{Y}} ||x - y||_2^2) \quad (4)$$

where σ_t denotes the threshold for clipping the distance at epoch t, calculated as $\sigma_t = max(10/t, 0.02)$ for epoch t.

In the training stage, we simultaneously optimized the registration decoder network parameters θ and latent vectors for all shape pairs in the training set. In the inference stage, we fixed the decoder network parameter and optimized the latent vector for each input pair of point sets to produce the desired rigid transformation.

3.5 Shape Completion

In this study, we designed a shape completion network to recover the full shapes from partial observations. The network enforces the latent code to represent the global representation of full shapes instead of partial ones and hence enhances the performance of the registration network. Our shape completion network was designed using an auto-decoder architecture based on the method presented in DeepSDF [55]. To represent a shape S, we attempt to learn a signed distance function (SDF) that, for any given spatial point p, outputs the distance from this point to the closest point on the surface of the shape S, i.e., $SDF(S, p) = s_p$. It is noteworthy that p does not need to be on shape S. The sign of the SDF value indicates whether the point is inside (negative) or outside (positive) of the input surface. Interested readers can refer to [55] for further details.

More specifically, for each target shape $\mathcal{Y}$, a latent code z was paired with the shape to represent its shape information. The SDF value at spatial location y_j can be generated using a deep feed-forward network $g_\beta(\cdot)$, as shown in Eq. (5). Fig. 3 illustrates the pipeline of our shape completion network.

$$s'_j = g_\beta(z, y_j) \quad (5)$$

In the training phase, our model attempts to minimize a predefined loss function over a batch of M training shapes with respect to all latent codes $\{z_i\}_{i=1}^M$ and the network parameters β, as follows:

$$\mathcal{L}_{com}(z, \mathcal{Y}) = \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{K} (\mathcal{L}(g_\beta(z_i, y_{ij}), s_{ij}) + \frac{1}{\sigma^2} ||z_i||_2^2) \quad (6)$$

where M denotes the number of training shapes in the batch, and K denotes the number of SDF samples for each shape. $\mathcal{L}(\cdot, \cdot)$ denotes the loss function for penalizing the predicted SDF value s'_{ij} and the ground truth SDF value s_{ij}. The second term was used to enforce the latent codes to be drawn from a prior Gaussian distribution. We used a clamped $L1$ loss for $\mathcal{L}$ with a threshold of 0.03. The point sampling strategy is explained in Section 4.2. At the inference time, we fixed the network parameters β; hence the shape code z for each shape $\mathcal{Y}$ can be estimated via maximum-a-posterior (MAP) estimation.

3.6 Optimization Strategy

We trained our shape completion and registration networks with the full shapes in the training set. In the training stage, we simultaneously optimized the parameters of our shape completion and point set registration networks, as well as the latent codes for all shapes. The optimal network parameters were generated as follows:

$$\theta^{optim}, \beta^{optim} = \arg\min_{\theta, \beta, \{z\}_{i=1}^M} (\mathcal{L}_{reg}(\mathcal{X}_{\mathcal{T}i}, \mathcal{Y}_i) + \lambda \mathcal{L}_{com}(z_i, \mathcal{Y}_i)) \quad (7)$$

where the first term denotes the point set registration loss and the second term denotes the shape completion loss, and λ denotes the hyper-parameter for balancing these two tasks. Without specific mention, we set λ to 0.1 in the following sections. After training on full shapes from the training set, the learned decoder network parameters θ and β can provide prior knowledge for the optimization of latent codes on unseen shapes.

During the model evaluation, given a pair of partial observations with source point set $\mathcal{X}_i$ and target point set $\mathcal{Y}_i$, we fixed the two decoder networks and optimized only the latent code of partial target shape $\mathcal{Y}_i$. In this regard, we randomly initialized a latent code z_i for the target point set and optimized it via MAP estimation, as follows:

$$z_i^{optim} = \arg\min_{z_i} (\mathcal{L}_{reg}(\mathcal{X}_{\mathcal{T}i}, \mathcal{Y}_i) + \lambda \mathcal{L}_{com}(z_i, \mathcal{Y}_i)) \quad (8)$$

After the optimization process, the optimal latent code can be used to perform shape completion for the target shape and predict the desired geometric transformation from the source point set to the target point set using equations (1) and (5).). It is noteworthy that it would be possible to split our model into a two-step process: 1) Optimize latent code z to obtain the optimal target shape reconstruction, and 2) optimize latent code z using the chamfer distance between the reconstructed target shape $\mathcal{Y}'$ and transformed source shape $\mathcal{X}_\mathcal{T}$. However, in our experiments (Section 5.2), we discovered that this two-step process did not result in better performance compared with our model using joint optimization.

4 Experiments

4.1 Experimental Dataset

We evaluated the proposed JCRNet method for partial registration on unaligned ModelNet40 [66] benchmark dataset. The ModelNet40 dataset contains 12,311 shapes of 40 object categories. We split this dataset into 9,843 shapes for training and 2,468 shapes for testing. We trained our model using full shapes from the training set and then evaluated the registration performance with the partial shapes from the test set.

To generate the data for training and model evaluation, we randomly sampled 1024 points for each shape. Subsequently, for each source shape $\mathcal{X}$, we generated the transformed shapes $\mathcal{Y}$ by applying a rigid transformation: the rotation matrix is characterized by three rotation angles along the x-, y-, and z-axes, where each value is uniformly sampled from $[0, 45]$ unit degree, and the translation is uniformly sampled from $[-0.5, 0.5]$. Finally, we simulated partial point sets of $\mathcal{Y}$ by randomly selecting a point in the unit space and retaining its 768 nearest neighbors.

4.2 Implementation Details

For the point set registration network, we used a sequential structure of C(256)-C(128)-M-FC(128)-FC(64)-FC(3), where the numbers indicate the number of output channels, C denotes 1-dimensional convolution layer with a kernel size of 1, FC denotes

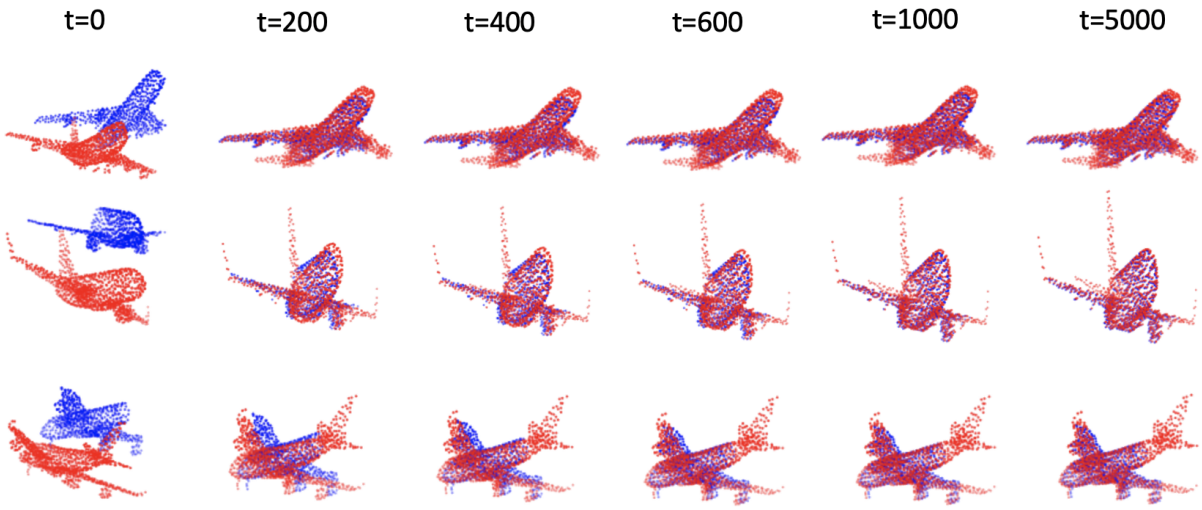

Fig. 4. Illustration of alignment process. The red points represent source point sets and the blue points represent the target point sets. We show the alignment results after 0, 200, 400, 600, 1000 and 5000 steps.

fully-connected layer, and M denotes max-pooling layer. To formulate a rigid transformation, our point set registration network predicts a 6-dimensional output for the desired rotation angles and translation vectors.

For the shape completion network, we leveraged the same architecture as that in [55] and used a 7-layer MLP to decode the latent code into SDF values, with each layer having a size of 512. To train the completion network, numerous SDF samples were constructed for each target shape. Each SDF sample included a 3D point and its corresponding SDF value. Specifically, we perturbed each point on the target shape with a zero-mean Gaussian distribution to generate three spatial samples, where the standard deviation was set to 0.2. For each point sample, we identified the nearest point from the original target shape and calculated the distance as its SDF value.

Our model was optimized using the Adam optimizer with an initial learning rate of 1e-3. We set the batch size to 50, momentum to 0.9, and weight decay to 1e-5. We used batch normalization and dropout for all fully connected layers, except for the output layer, both in our shape completion and point set registration networks. Each latent vector z was randomly initialized from a Gaussian distribution $\mathcal{N}(0, 0.06)$ with a size of 256. The proposed method was implemented using the PyTorch library on a TESLA K80 GPU.

We evaluated the point set registration performance using three metrics, including mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE). Angular measurements were recorded in units of degrees. To demonstrate the effectiveness of the proposed JCRNet, we compared the performance of our model with those of prevalent learning-based methods, including DCP [1], PR-Net [5], and DGR [67]. For DGR, the feature extractor was pre-trained following [68] and then the whole model was finetuned for point set registration, with a voxel size of 0.01. Other settings were the same as [67]. For all the methods compared, i.e., DCP, PR-Net, and DGR, the models were trained on all 40 object categories. However, in our method, to facilitate the training of the shape completion network, we conducted experiments for each object category independently.

4.3 Transformation Process

Before presenting the experimental results, we examine the alignment process of three test pairs by the optimization steps. As shown in Fig. 4, the proposed model aligned the main components of the input shapes after 600 optimization steps. Although the rotation predictions were close to the ground truth, thousands of optimization steps were still required by our model to obtain accurate translation predictions. After 1000 steps, our model yielded a satisfactory alignment for all the input shape pairs.

4.4 Partial-to-Full Registration Results

We first evaluated the performance of our JCRNet on the ModelNet40 dataset. Table 1 lists the performance of our method and the compared methods on four categories (airplane, bench, chair, and sofa) of the ModelNet40 dataset. As shown in Table 1, our model performed better than all the compared methods for rotation prediction and was an order of magnitude better than DCP and PR-Net. For the rotation prediction, our JCRNet achieved an RMSE(R) of less than 1° for all four classes. The DGR method demonstrated the second-best registration performance; however, its feature encoder network required pre-training based on the method in [68]. For the translation prediction, our JCRNet model got better or comparable performance with another self-supervised method PR-Net. The two supervised methods, i.e., DCP and DGR, obtained better performance for translation prediction because they use ground truth translation labels for supervision.

It is noteworthy that the compared methods, including DCP, PR-Net, and DGR, solve the geometric transformation based on dense point correspondences between the transformed source and target shapes. Nevertheless, it is nontrivial to identify dense point correspondences using point-wise features, particularly for partial observations. By contrast, our JCRNet model does not use point correspondence information but directly regresses the geometric transformation using fully connected layers. Therefore,

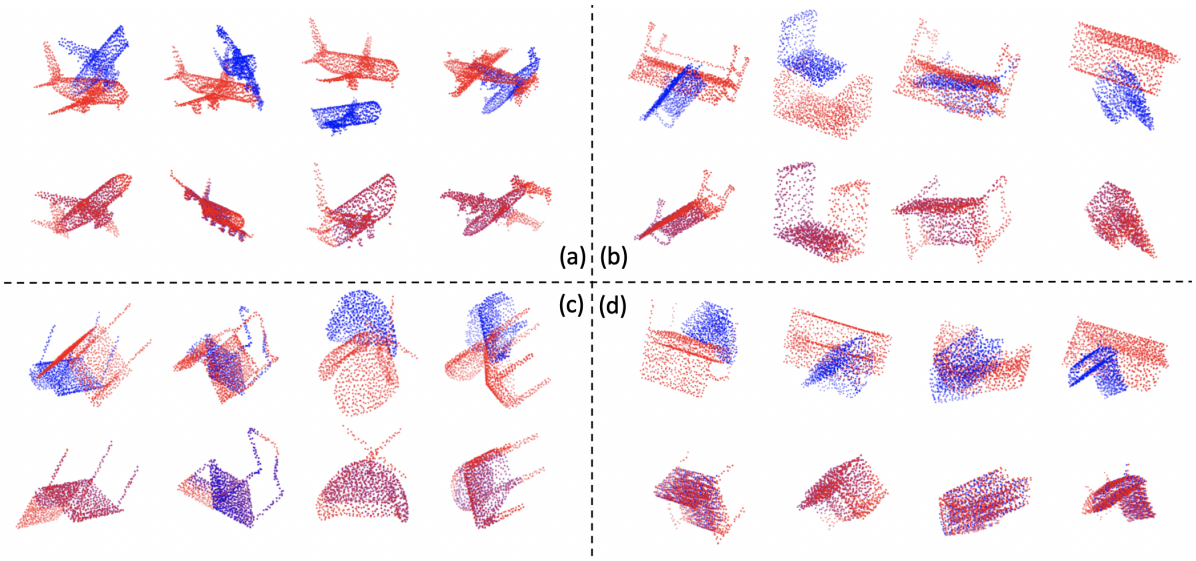

Fig. 5. Randomly selected examples of partial point set registration on different categories, (a) airplane, (b) bench, (c) chair and (d) sofa. The red points represent source point sets and the blue points represent the target point sets. The odd rows show input shapes, and the even rows show output results.

Model	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
DCP [1]	20.518583	4.529744	3.281942	0.001546	0.039315	0.029034
PRNet [5]	6.349841	2.519889	1.926538	0.002804	0.052949	0.041235
DGR [67]	0.774078	0.879817	0.559655	**0.000070**	**0.008366**	**0.005476**
JCRNet (Ours)	**0.058920**	**0.228857**	**0.138904**	0.003276	0.057238	0.019733
DCP [1]	13.646673	3.694140	2.743233	0.002339	0.048364	0.036228
PRNet [5]	9.687140	3.112417	2.276538	0.003241	0.056932	0.044811
DGR [67]	0.986714	0.993335	0.662674	**0.000317**	**0.017807**	**0.014062**
JCRNet (Ours)	**0.049004**	**0.214932**	**0.151105**	0.001628	0.040348	0.010773
DCP [1]	35.288406	5.940404	4.494213	0.001378	0.037116	0.029175
PRNet [5]	13.405939	3.661412	2.761630	0.001313	0.036239	0.029226
DGR [67]	5.475403	2.339958	1.019728	0.000078	0.008860	0.005168
JCRNet (Ours)	**0.214308**	**0.266104**	**0.123464**	**0.000074**	**0.007140**	**0.003060**
DCP [1]	10.668366	3.266247	2.458252	0.000892	0.029860	0.022282
PRNet [5]	5.853882	2.419480	1.819235	0.001110	0.033323	0.025116
DGR [67]	0.262347	0.512198	0.339514	**0.000068**	**0.008226**	**0.006084**
JCRNet (Ours)	**0.179997**	**0.408753**	**0.287125**	0.001675	0.039729	0.018096

TABLE 1
From top to bottom: test performance on the airplane, bench, chair and sofa categories. Boldface indicates the best performance.

our model can be trained in a completely unsupervised manner and is sufficiently flexible to be equipped in different network architectures. Randomly selected examples are presented in Fig. 5. As shown, our model can align partial inputs with different orientations in all four categories. We omitted the results of other categories because our model was trained independently for each category. Conducting experiments on all 40 categories of the ModelNet40 dataset would be time-consuming.

4.5 Partial-to-Partial Registration Results

To further demonstrate the effectiveness of our method in a more realistic scenario, we evaluated the performance of our model in a partial-to-partial setting. In this regard, we simulated partial point sets for both the source and target shapes following the settings of PR-Net [5]. First, we randomly sampled a point in the unit space and then selected its 768 nearest neighbors for both the source and target point sets. We conducted experiments on four categories (airplane, bench, chair, and sofa) from the ModelNet40 dataset. Quantitative results are shown in Table 2. As shown in Table 2, our JCRNet model performed significantly better than DCP and PR-Net for all object categories for rotation prediction. In addition, our model achieved better rotation prediction than DGR for all categories except the sofa category. It is noteworthy that DGR requires its feature encoder network to be pre-trained based on the method in [68] to ensure high performance, whereas our JCRNet model is trained in an end-to-end manner. Our JCRNet model achieved the best performance for translation prediction in two of the four categories. It is noteworthy that the point dropping strategy used in [5] maintains a significant portion of the overlap for the sampled partial point sets. Hence, all methods except DCP achieve better performances under the partial-to-partial setting than under the partial-to-full setting. Randomly selected visualization results are shown in Fig. 6.

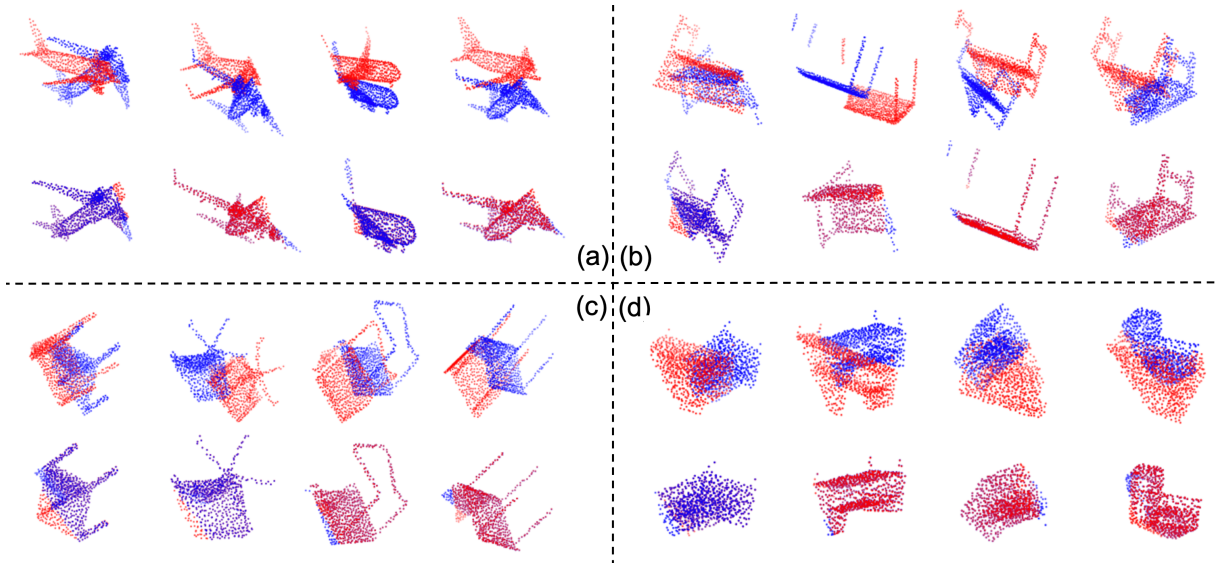

Fig. 6. Randomly selected examples of partial-to-partial point set registration on different categories, (a) airplane, (b) bench, (c) chair and (d) sofa. The red points represent source point sets and the blue points represent the target point sets. The odd rows show input shapes, and the even rows show output results.

Model	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
DCP [1]	18.777311	4.333280	3.145719	0.006297	0.079350	0.061786
PRNet [5]	1.172203	1.082683	0.705065	0.000338	0.018398	0.012744
DGR [67]	0.377046	0.614041	0.331919	0.000046	0.006757	0.003316
JCRNet (Ours)	**0.010427**	**0.102114**	**0.064069**	**0.000026**	**0.005141**	**0.001446**
DCP [1]	29.434374	5.425345	4.428350	0.008292	0.091062	0.066240
PRNet [5]	0.652327	0.807667	0.543816	0.000197	0.014035	0.009289
DGR [67]	0.230931	0.480553	0.299004	0.000034	0.005843	0.003631
JCRNet (Ours)	**0.025572**	**0.159913**	**0.095813**	**0.002307**	**0.048027**	**0.016239**
DCP [1]	54.692078	7.395409	5.696293	0.004645	0.068157	0.055036
PRNet [5]	5.199563	2.280255	1.282853	0.000256	0.016000	0.010545
DGR [67]	1.419951	1.191617	0.634732	0.000027	0.005188	0.003001
JCRNet (Ours)	**0.231249**	**0.480884**	**0.146714**	**0.000001**	**0.001152**	**0.000386**
DCP [1]	17.577272	4.192526	3.132129	0.007614	0.087256	0.069179
PRNet [5]	1.074889	1.036769	0.707419	0.000193	0.013890	0.008943
DGR [67]	**0.030929**	**0.175867**	**0.116051**	**0.000005**	**0.002214**	**0.001395**
JCRNet (Ours)	0.045184	0.212564	0.126405	0.000172	0.013132	0.004499

TABLE 2
From top to bottom: partial-to-partial registration performance on the airplane, bench, chair, and sofa categories respectively. Boldface indicates the best performance.

Model	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
DCP [1]	27.383821	5.232955	3.716614	0.002059	0.045373	0.034398
PR-Net [5]	7.609180	2.758474	2.119697	0.003375	0.058099	0.045531
DGR [67]	22.786913	4.773564	3.289794	0.012247	0.110665	0.093175
JCRNet (Ours)	**3.540532**	**1.273092**	**0.376848**	**0.000425**	**0.017153**	**0.005731**
DCP [1]	18.817074	4.337865	3.29574	0.004637	0.068093	0.049714
PR-Net [5]	9.868777	3.141461	2.333664	0.003426	0.058536	0.045699
DGR [67]	25.916723	5.090847	4.090961	0.026187	0.161824	0.138757
JCRNet (Ours)	**0.058920**	**0.228857**	**0.138904**	**0.003276**	**0.057238**	**0.019733**
DCP [1]	48.763847	6.983111	5.391983	0.00156	0.039496	0.031935
PR-Net [5]	13.439675	3.666016	2.746882	0.001359	0.036871	0.030141
DGR [67]	39.876206	6.314761	4.06904	0.001863	0.043159	0.034006
JCRNet (Ours)	**0.305641**	**0.406611**	**0.198128**	**0.000040**	**0.005002**	**0.002327**
DCP [1]	14.932157	3.864215	2.872617	0.001026	0.032031	0.024701
PR-Net [5]	6.238619	2.497723	1.836021	0.001127	0.033567	0.025541
DGR [67]	17.028861	4.126604	2.82924	0.004347	0.065929	0.055279
JCRNet (Ours)	**0.261543**	**0.484324**	**0.331687**	**0.002196**	**0.044315**	**0.024815**

TABLE 3
From top to bottom: test on unseen point clouds on airplane, bench, chair and sofa categories with Gaussian noise ($\sigma = 0.01$). Boldface indicates the best performance.

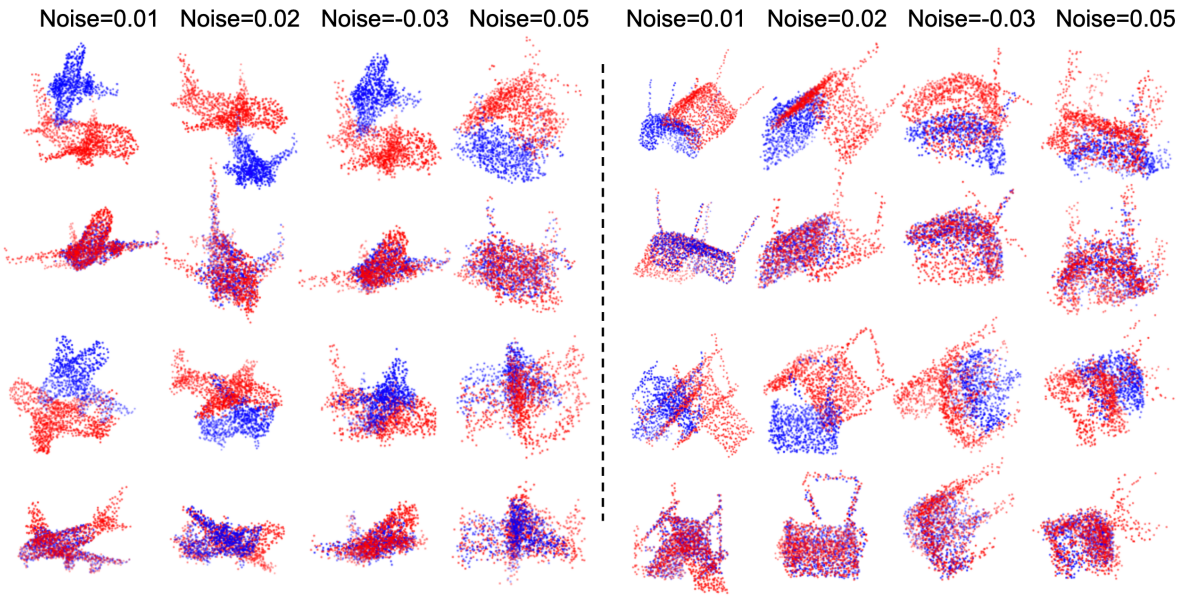

Fig. 7. Visualization of the registration results with Gaussian noise on airplane (left) and chair (right) categories. The odd rows show input shapes and the even rows show output results.

5 Discussions

5.1 Robustness Test

5.1.1 Resilience to Gaussian Noise

We further evaluated the robustness of the proposed model under Gaussian noise. In this regard, we perturbed each point on both source and target point sets with random noise, which was sampled from $\mathcal{N}(0, \sigma)$ and clipped to $[-5\sigma, +5\sigma]$. Other settings were the same as those listed in Section 4.2. We trained our JCRNet model and the compared methods, including DCP, PR-Net, and DGR, on noise-free data and evaluated the performance of all methods on the test set with Gaussian noise. Table 3 shows the registration performance with σ set to 0.01. As shown in this table, our model achieved significantly better performance than all compared methods for both rotation and translation prediction. By comparing Table 1 and Table 3, it was clear that the performance of the DCP and DGR models degraded significantly, with RMSE(R) increasing from 3.266247° and 0.512198° to 3.864215° and 4.126604° on the sofa category, respectively. Although DGR achieved the second-best performance on the noise-free dataset for partial point set registration, it performed even worse than DCP and PR-Net on point sets with Gaussian noise. By contrast, the performance of two self-supervised methods, including our JCRNet model and PR-Net [5], decreased much smaller, with RMSE(R) increasing from 2.419480° and 0.408753° to 2.497723° and 0.484324°, respectively. This demonstrates that compared to the DCP and DGR models, our model and PR-Net were more robust to data noise for partial point set registration. Fig. 7 shows selected examples of our model on airplane and chair categories with Gaussian noise ($\sigma = 0.01$). As shown in Fig. 7, even when the input point sets contained Gaussian noise, our model successfully aligned the source and target point sets. Fig. 8 shows the performance of our model and compared methods with different noise levels, as indicated by RMSE(R) and RMSE(t). As shown in this figure, our model achieved better performance than all comparing methods under various noise levels. Specifically, with the increase of Gaussian noise levels, the performance of our JCRNet model degraded slowly, whereas the performance of the compared methods degraded significantly, particularly the DGR model.

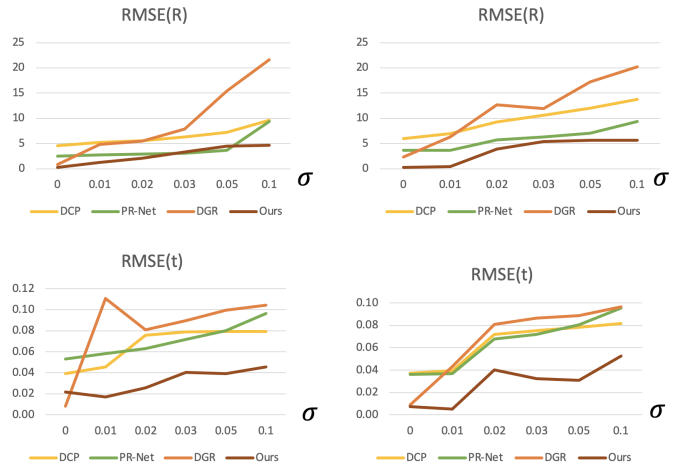

Fig. 8. Test performance with different levels of Gaussian noise on airplane (left) and chair (right) categories.

5.1.2 Resilience to Outliers

We further evaluated the performance of our model on point sets with outliers. In this regard, we added outlier points to the target point sets. The other settings were the same as those listed in Section 4.2. We trained our JCRNet model and the compared methods, including DCP, PR-Net, and DGR, on noise-free data, and evaluated the performance of all methods on the test set with outliers. Table 4 lists the registration performance of the comparison methods with an outlier ratio of 0.1. As shown in this table, our model performed significantly better than all the compared methods for rotation prediction. This shows that our JCRNet model is more robust to outlier points than the current state-of-the-art DGR model. It is noteworthy that the DGR model

performed better than our JCRNet model for translation prediction in three out of four categories. This is because the DGR model uses ground truth translation labels as supervision, whereas JCRNet is optimized via an unsupervised chamfer loss. Fig. 9 shows selected examples of our model on the chair category with different outlier ratios. As shown in Fig. 9, even when the input point sets contained outliers, our model successfully aligned the source and target point sets.

5.1.3 Robustness of Missing Points

We explored the performance of our model with different numbers of missing points. We conducted experiments on the airplane and chair categories with missing ratios of 25%, 37.5%, and 50%, i.e., each partial shape has 768, 640, and 512 remaining points. Table 5 shows the quantitative results of our model. As shown in Table 5, as the number of missing points increased, the performance of our model deteriorated, as expected. For the partial shapes with 640 remaining points, our model achieved an RMSE(R) of approximately 1° and an RMSE(t) of less than 0.06. Even with 512 missing points, our model maintained an RMSE(R) of less than 5° for both airplane and chair categories. We also noticed that our model got large RMSE(R) values when the input shapes have 512 missing points. From our experiments, we observed that our model failed to align several cases in which the rotation predictions had large errors. This is because, when numerous points are missing in the input shapes, our model mighty converges to a local minimum, considering that the latent code z was randomly initialized from a Gaussian distribution. These failure predictions dominated the prediction errors, particularly for MSE(R).

Fig. 10 shows selected examples of the registration results of our model on airplane and chair categories with missing ratios of 37.5% and 50%, respectively. As shown in Fig. 10, our model yielded satisfactory alignment in most cases. The cases shown in the blue boxes indicate failure cases. For these failure cases, our model predicted a rotation matrix with large errors. In our experiments, we discovered that our model failed in only a few cases where the missing ratio was less than 50%.

5.2 Ablation Analysis

In this section, we conducted further experiments to demonstrate the effectiveness of the shape completion module. For this experiment, we removed the shape completion component from our method, i.e., we set λ to 0. Hence, our method directly performs point set registration for the partial point sets. We mark this model as the "baseline 1" model. Moreover, to demonstrate the superiority of joint training of the shape completion and registration networks instead of training them separately, we conducted experiments by running these two networks separately and mark this model as the ''baseline 2" model. Table 6 lists the performances of our model, the two baseline models, and the compared methods, i.e., DCP [1], PR-Net [5], and DGR [67]. As shown in the table, both our model and the two baseline models performed significantly better performance than the DCP and PR-Net models. By comparing ''baseline 1" and "baseline 2', it is clear that using the shape completion module can improve the performance of partial point set registration. By comparing the last two rows, it is evident that our model achieved significant performance improvement through the joint training of the two networks, particularly for rotation prediction.

5.3 Inference Time Comparison

In this section, we compare the inference time of our method with those of DCP, PR-Net and DGR. We calculated the average operating time for all test shapes from the ModelNet40 dataset. The experiments were conducted on two Titan XP GPUs with a batch size of 50. Table 7 shows the inference time of the compared methods. Our JCTNet required 12.11 s to optimize 3000 steps until convergence for each pair of shapes. Meanwhile, the PR-Net required self-supervised optimization. In fact, it required 37.5 h to be optimized for 100 epochs on 2468 test shapes. Therefore, the registration of each pair of shapes required 54.7 s on average. Under a self-supervised setting, our JCRNet was much faster than the PR-Net model. DCP and DGR required less than 1 s for the registration of each pair, which was much faster than our JCRNet and PR-Net. This is because the DCP and DGR are one-shot-based methods that can directly predict the registration results in one network forward pass.

5.4 Shape Completion Results

In this section, we show that our model can yield satisfactory shape completion/reconstruction results. We report the shape completion performance on four categories (airplane, bench, chair, and sofa) of the ModelNet40 dataset and compare the performance of our JCRNet model with those of state-of-the-art methods, including PCN [56], AtlasNet [69], and CRN [70]. For a fair comparison, we sampled the partial point sets following Section 4.1 and performed data augmentation for all the compared methods. As shown in Table 8, the performance of our method was comparable to those of state-of-the-art methods. In particular, our model performed better than PCN and AtlasNet on three out of four object categories. It is noteworthy that all the compared methods only focused on shape completion, whereas our JCRNet addressed a more challenging task that simultaneously achieves shape completion and registration. Moreover, the best-performing method, CRN [70], uses adversarial loss to yield more realistic reconstruction results, whereas other methods do not involve adversarial constraints.

Fig. 11 shows the randomly selected results in the airplane category. We generated grid points in a unit space with a spatial resolution of 0.25 and generated 40 40 40 points for each shape. We input these points to the shape completion network and predicted their SDF values. Only points with an SDF value less than 0.015 were retained for visualization. As shown in Fig. 11, our JCRNet model can potentially recover full objects from partial observations.

6 Conclusion and Limitations

In this paper, we introduced an unsupervised method for category-specific partial point set registration. Since recent learning-based methods have achieved significantly better registration performance on the full shapes instead of on partial observations, we suggested bridging the performance gaps by incorporating a shape completion network to recover full shapes from partial observations. Hence, we introduced a learnable latent code for each target shape, which can be regarded as the global feature encoding of the target shape. This latent code was initialized from a Gaussian distribution and was used as the input for both shape completion and registration networks. In this way, our model obviated the necessity for the explicit design of a point feature encoding network and, more importantly, enabled the joint training of

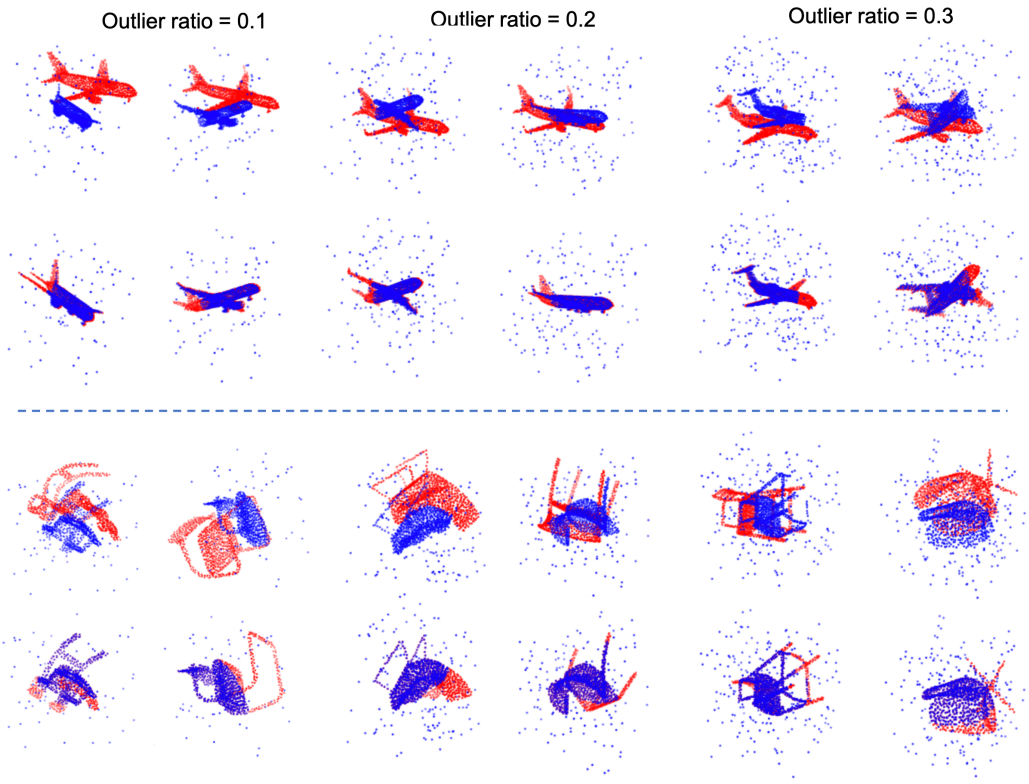

Fig. 9. Visualization of the registration results on chair category with an outlier ratio of 0.1, 0.2 and 0.3. Odd rows show input point sets and even rows show output results.

Model	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
DCP [1]	39.577682	6.291080	4.865984	0.006664	0.081633	0.063203
PR-Net [5]	178.689117	13.367465	6.018843	0.004905	0.070034	0.053326
DGR [67]	0.764659	0.874448	0.558550	**0.000069**	**0.008308**	**0.005432**
JCRNet (Ours)	**0.105959**	**0.325513**	**0.235938**	0.000484	0.021999	0.017848
DDCP [1]	41.437172	6.437171	5.062754	0.008964	0.094680	0.071456
PR-Net [5]	72.204147	8.497302	5.662551	0.006656	0.081586	0.068024
DGR [67]	1.128649	1.062379	0.690827	**0.000314**	**0.017720**	**0.013945**
JCRNet (Ours)	**0.315161**	**0.561392**	**0.376345**	0.002127	0.046121	0.025024
DCP [1]	69.053513	8.309844	6.376768	0.004785	0.069177	0.056092
PR-Net [5]	60.598633	7.784513	5.985467	0.004487	0.066983	0.052878
DGR [67]	4.311404	2.076392	0.985787	**0.000073**	**0.008524**	**0.005170**
JCRNet (Ours)	**1.251415**	**1.118667**	**0.251478**	0.001239	0.035202	0.007975
DCP [1]	24.780735	4.978025	3.815879	0.008009	0.089491	0.071516
PR-Net [5]	17.705265	4.207762	3.234045	0.003873	0.062234	0.050455
DGR [67]	17.394764	4.170703	2.900578	0.018805	0.137133	0.114736
JCRNet (Ours)	**0.383707**	**0.619441**	**0.333120**	**0.003504**	**0.059199**	**0.032810**

TABLE 4
From top to bottom: test performance on unseen point clouds with an outlier ratio of 0.1 on airplane, bench, chair and sofa categories. Boldface indicates the best performance.

Missing ratio	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
25%	0.058920	0.228857	0.138904	0.003276	0.057238	0.019733
37.5%	1.262889	0.992460	0.577019	0.000553	0.023497	0.019172
50%	19.755819	3.851332	1.201197	0.003392	0.057192	0.021514
25%	0.214308	0.266104	0.123464	0.000074	0.007140	0.003060
37.5%	3.638625	1.242836	0.452099	0.000169	0.012908	0.005759
50%	20.928379	4.574755	0.994808	0.060251	0.245461	0.204531

TABLE 5
Test performance with different missing ratios on airplane (top) and chair (bottom) categories. Missing ratio indicates the ratio of missing points with respect to full observations. Boldface indicates the best performance.

both shape completion and registration networks, thereby affording performance improvement through inter-enhancement. During training, the latent code was optimized simultaneously with the parameters of the shape completion and registration networks.

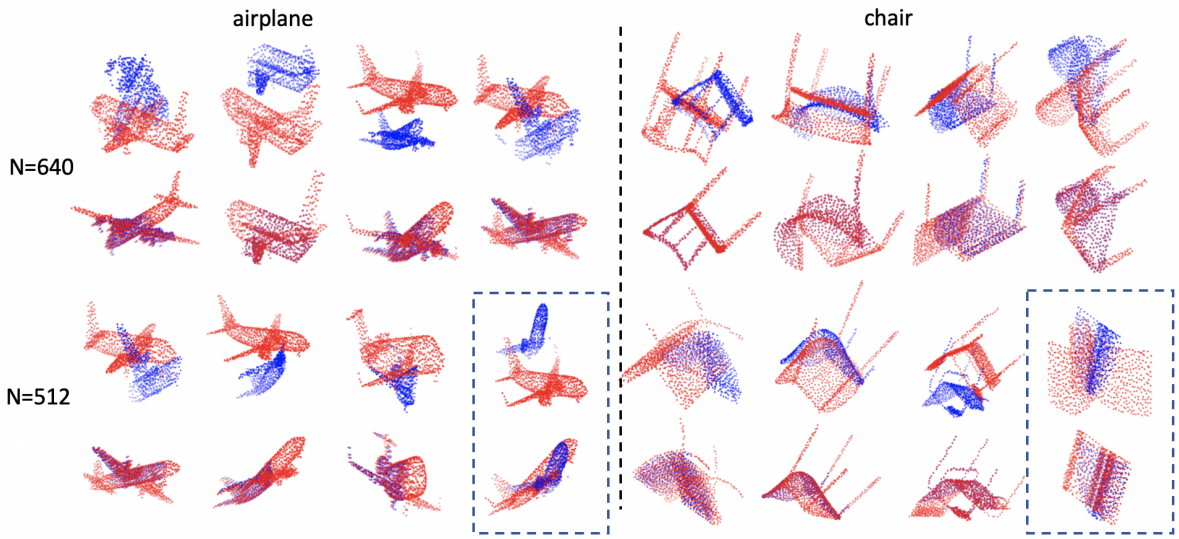

Fig. 10. Visualization of different number of missing points on airplane (left) and chair (right) categories. The odd rows show input shapes, and the even rows show output results. Blue boxes show failure cases.

Model	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
DCP [1]	20.518583	4.529744	3.281942	0.001546	0.039315	0.029034
PRNet [5]	6.349841	2.519889	1.926538	0.002804	0.052949	0.041235
DGR [67]	0.774078	0.879817	0.559655	0.000070	0.008366	0.005476
Ours (baseline 1)	2.202928	1.394375	0.522311	0.003366	0.057668	0.024253
Ours (baseline 2)	1.480211	1.171062	0.661237	0.003017	0.054924	0.021381
Ours ($\lambda = 0.1$)	**0.058920**	**0.228857**	**0.138904**	**0.003276**	**0.057238**	**0.019733**

TABLE 6

Ablation analysis: test performance on airplane category of ModelNet40 dataset. The 'baseline 1' is our model without the shape completion network, and 'baseline 2' is our model that performs shape completion and registration separately. Boldface indicates the best performance.

Method	DCP [1]	PR-Net [5]	DGR [67]	Ours
Time	**0.03s**	54.70s	0.04s	12.11s

TABLE 7

Inference Time of different methods.

Method	Airplane	Bench	Chair	Sofa
PCN [56]	0.0272	0.0356	0.0383	0.0410
AtlasNet [69]	0.0324	0.0431	0.0452	0.0451
CRN [70]	**0.0174**	**0.0212**	**0.0257**	**0.0244**
Ours	<u>0.0204</u>	<u>0.0301</u>	<u>0.0508</u>	<u>0.0342</u>

TABLE 8

Shape completion performance on four categories (airplane, bench, chair and sofa) of ModelNet40 dataset. Boldface indicates the best performance and underline indicates the second best performance.

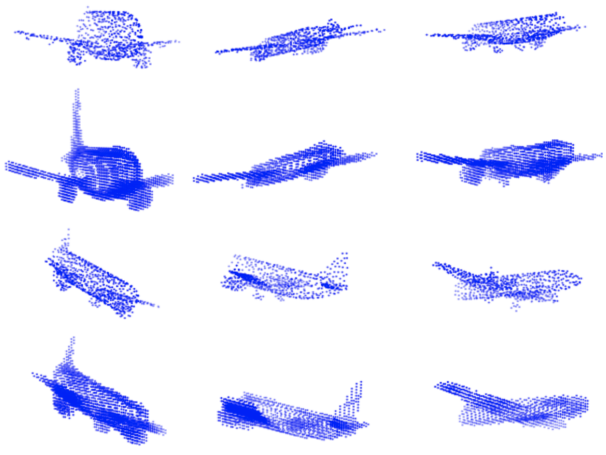

Fig. 11. Selected examples of shape completion results. Odd rows show partial observations, and even rows show reconstruction results.

In the inference stage, the network parameters were fixed and used as prior knowledge for guiding the optimization of the latent codes to obtain the optimal shape completion and registration results. Experiments on the ModelNet40 dataset demonstrated the effectiveness of our model for rigid transformation prediction based on partial observations. Additionally, the results showed that our model is robust to input noise, outliers, and missing points.

The proposed method, in its current form, can only be used for seen categories, as the shape completion network requires per-category training. This limitation arises because our registration network will benefit less from shape completion on unseen categories, since our completion network in its current form will demonstrate degraded performance on unseen categories. Hence, our proposed method was trained and evaluated by category; furthermore, it is not intended for use in cross-category partial shape registration. Additionally, the proposed method, in its current form, primarily focuses on modeled data from seen categories and is currently not evaluated based on scans acquired from partial views of 3D scenes/models.

7 ACKNOWLEDGEMENT

This work was supported by the NYU Abu Dhabi Institute (AD131).

REFERENCES

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

[2] 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. International Society for Optics and Photonics, 1992, pp. 586–607.

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

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

[5] Y. Wang and J. M. Solomon, "Prnet: Self-supervised learning for partial-to-partial registration," Advances in neural information processing systems, 2019.

[6] B. PaulJ and M. NeilD, "A method for registration of 3-d shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239–256, 1992.

[7] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in Proceedings third international conference on 3-D digital imaging and modeling. IEEE, 2001, pp. 145–152.

[8] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in Robotics: science and systems, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[9] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in Computer graphics forum, vol. 32, no. 5. Wiley Online Library, 2013, pp. 113–123.

[10] G. Agamennoni, S. Fontana, R. Y. Siegwart, and D. G. Sorrenti, "Point clouds registration with probabilistic data association," in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016, pp. 4092–4098.

[11] T. Hinzmann, T. Stastny, G. Conte, P. Doherty, P. Rudol, M. Wzorek, E. Galceran, R. Siegwart, and I. Gilitschenski, "Collaborative 3d reconstruction using heterogeneous uavs: System and experiments," in International Symposium on Experimental Robotics. Springer, 2016, pp. 43–56.

[12] 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.

[13] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "An unsupervised learning model for deformable medical image registration," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9252–9260.

[14] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," in Symposium on geometry processing, vol. 2, no. 3. Vienna, Austria, 2005, p. 5.

[15] S. Krishnan, P. Y. Lee, J. B. Moore, S. Venkatasubramanian et al., "Global registration of multiple 3d point sets via optimization-on-a-manifold." in Symposium on Geometry Processing, 2005, pp. 187–196.

[16] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas, "Registration of point cloud data from a geometric optimization perspective," in Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, 2004, pp. 22–31.

[17] O. Litany, A. M. Bronstein, and M. M. Bronstein, "Putting the pieces together: Regularized multi-part shape matching," in European Conference on Computer Vision. Springer, 2012, pp. 1–11.

[18] H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching," in Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, vol. 2. IEEE, 2000, pp. 44–51.

[19] A. Myronenko, X. Song, and M. A. Carreira-Perpinán, "Non-rigid point set registration: Coherent point drift," in Advances in Neural Information Processing Systems, 2007, pp. 1009–1016.

[20] O. Halimi, I. Imanuel, O. Litany, G. Trappolini, E. Rodolà, L. Guibas, and R. Kimmel, "The whole is greater than the sum of its nonrigid parts," arXiv preprint arXiv:2001.09650, 2020.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[24] X. Li, X. Yao, and Y. Fang, "Building-a-nets: robust building extraction from high-resolution remote sensing images with adversarial networks," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 11, no. 10, pp. 3680–3687, 2018.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91–99.

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[27] Y. Hu, X. Li, N. Zhou, L. Yang, L. Peng, and S. Xiao, "A sample update-based convolutional neural network framework for object detection in large-area remote sensing images," IEEE Geoscience and Remote Sensing Letters, vol. 16, no. 6, pp. 947–951, 2019.

[28] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 6148–6157.

[29] J. Chen, L. Wang, X. Li, and Y. Fang, "Arbicon-net: Arbitrary continuous geometric transformation networks for image registration," in Advances in Neural Information Processing Systems, 2019, pp. 3415–3425.

[30] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, vol. 1, no. 2, p. 4, 2017.

[31] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in Proceedings of the IEEE international conference on computer vision workshops, 2015, pp. 37–45.

[32] X. Li, L. Wang, M. Wang, C. Wen, and Y. Fang, "Dance-net: Density-aware convolution networks with context encoding for airborne lidar point cloud classification," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 166, pp. 128–139, 2020.

[33] L. Wang, X. Li, and Y. Fang, "Few-shot learning of part-specific probability space for 3d shape segmentation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4504–4513.

[34] X. Li, C. Wen, L. Wang, and Y. Fang, "Topology-constrained shape correspondence," IEEE transactions on visualization and computer graphics, 2020.

[35] L. Wang, J. Chen, X. Li, and Y. Fang, "Non-rigid point set registration networks," arXiv preprint arXiv:1904.01428, 2019.

[36] X. Liu, C. R. Qi, and L. J. Guibas, "Flownet3d: Learning scene flow in 3d point clouds," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 529–537.

[37] 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.

[38] O. Sorkine and D. Cohen-Or, "Least-squares meshes," in Proceedings Shape Modeling Applications, 2004. IEEE, 2004, pp. 191–199.

[39] W. Zhao, S. Gao, and H. Lin, "A robust hole-filling algorithm for triangular mesh," The Visual Computer, vol. 23, no. 12, pp. 987–997, 2007.

[40] S. Thrun and B. Wegbreit, "Shape from symmetry," in Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol. 2. IEEE, 2005, pp. 1824–1831.

[41] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas, "Discovering structural regularity in 3d geometry," in ACM SIGGRAPH 2008 papers, 2008, pp. 1–11.

[42] V. Kraevoy and A. Sheffer, "Template-based mesh completion." in Symposium on Geometry Processing, vol. 385. Citeseer, 2005, pp. 13–22.

[43] O. Litany, A. Bronstein, M. Bronstein, and A. Makadia, "Deformable shape completion with graph convolutional autoencoders," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[44] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in Proceedings of the 26th annual conference on Computer graphics and interactive techniques, 1999, pp. 187–194.

[45] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," ACM transactions on graphics (TOG), vol. 30, no. 4, pp. 1–10, 2011.

[46] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: shape completion and animation of people," in ACM SIGGRAPH 2005 Papers, 2005, pp. 408–416.

[47] A. Weiss, D. Hirshberg, and M. J. Black, "Home 3d body scans from noisy image and range data," in 2011 International Conference on Computer Vision. IEEE, 2011, pp. 1951–1958.

[48] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem, "Completing 3d object shape from one depth image," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2484–2493.

[49] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in European conference on computer vision. Springer, 2016, pp. 628–644.

[50] J. Wu, C. Zhang, X. Zhang, Z. Zhang, W. T. Freeman, and J. B. Tenenbaum, "Learning shape priors for single-view 3d completion and reconstruction," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 646–662.

[51] A. Dai, C. Ruizhongtai Qi, and M. Nießner, "Shape completion using 3d-encoder-predictor cnns and shape synthesis," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5868–5877.

[52] D. Stutz and A. Geiger, "Learning 3d shape completion from laser scan data with weak supervision," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1955–1964.

[53] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in Advances in neural information processing systems, 2016, pp. 82–90.

[54] Q. Tan, L. Gao, Y.-K. Lai, and S. Xia, "Variational autoencoders for deforming 3d mesh models," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5841–5850.

[55] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 165–174.

[56] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network," in 2018 International Conference on 3D Vision (3DV). IEEE, 2018, pp. 728–737.

[57] X. Wen, T. Li, Z. Han, and Y.-S. Liu, "Point cloud completion by skip-attention network with hierarchical folding," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1939–1948.

[58] M. Sarmad, H. J. Lee, and Y. M. Kim, "Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5898–5907.

[59] S. Gurumurthy and S. Agrawal, "High fidelity semantic shape completion for point clouds using latent optimization," in 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2019, pp. 1099–1108.

[60] O. Litany, A. Bronstein, M. Bronstein, and A. Makadia, "Deformable shape completion with graph convolutional autoencoders," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1886–1895.

[61] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4460–4470.

[62] S. Tan and M. L. Mayrovouniotis, "Reducing data dimensionality through optimizing neural network inputs," AIChE Journal, vol. 41, no. 6, pp. 1471–1480, 1995. [Online]. Available: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690410612

[63] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, "Meta-learning with latent embedding optimization," arXiv preprint arXiv:1807.05960, 2018.

[64] M. Bouakkaz and M.-F. Harkat, "Combined input training and radial basis function neural networks based nonlinear principal components analysis model applied for process monitoring." in IJCCI, 2012, pp. 483–492.

[65] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "3d-coded: 3d correspondences by deep deformation," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 230–246.

[66] 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.

[67] 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.

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

[69] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3d surface generation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 216–224.

[70] X. Wang, M. H. Ang Jr, and G. H. Lee, "Cascaded refinement network for point cloud completion," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 790–799.

Xiang Li received a B.S. degree in remote sensing science and technology from Wuhan University, Wuhan, China, in 2014. He received a Ph.D. in cartography and GIS from the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China, in 2019. He is currently a Postdoctoral Associate with the Department of Electrical and Computer Engineering, New York University Abu Dhabi, Abu Dhabi, United Arab Emirates. His research interests include computer vision, photogrammetry, and remote sensing.

Lingjing Wang received a B.S. degree from Moscow State University, Moscow, Russia, in 2011. He received a Ph.D. at the Courant Institute of Mathematical Science, New York University, USA, in 2019. He is currently a Postdoctoral Associate with the Department of Electrical and Computer Engineering, New York University Abu Dhabi, Abu Dhabi, United Arab Emirates. His research interests include deep learning and 3D visual computing.

Yi Fang received B.S. and M.S. degrees in biomedical engineering from Xi'an Jiaotong University, Xi'an, China, in 2003 and 2006, respectively, and a Ph.D. in mechanical engineering from Purdue University, West Lafayette, IN, USA, in 2011. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, New York University Abu Dhabi, Abu Dhabi, United Arab Emirates. His research interests include three-dimensional computer vision and pattern recognition, large-scale visual computing, deep visual computing, deep cross-domain and cross-modality multimedia analysis, and computational structural biology.