SIRA-PCR: Sim-to-Real Adaptation for 3D Point Cloud Registration

Suyi Chen[1,4*] Hao Xu[2*] Ru Li[3] Guanghui Liu[1†] Chi-Wing Fu[2] Shuaicheng Liu[1,4†]

[1]University of Electronic Science and Technology of China
[2]The Chinese University of Hong Kong [3]Harbin Institute of Technology [4]Megvii Technology

Abstract

Point cloud registration is essential for many applications. However, existing real datasets require extremely tedious and costly annotations, yet may not provide accurate camera poses. For the synthetic datasets, they are mainly object-level, so the trained models may not generalize well to real scenes. We design SIRA-PCR, a new approach to 3D point cloud registration. First, we build a synthetic scene-level 3D registration dataset, specifically designed with physically-based and random strategies to arrange diverse objects. Second, we account for variations in different sensing mechanisms and layout placements, then formulate a sim-to-real adaptation framework with an adaptive re-sample module to simulate patterns in real point clouds. To our best knowledge, this is the first work that explores sim-to-real adaptation for point cloud registration. Extensive experiments show the SOTA performance of SIRA-PCR on widely-used indoor and outdoor datasets. The code and dataset will be released on https://github.com/Chen-Suyi/SIRA_Pytorch.

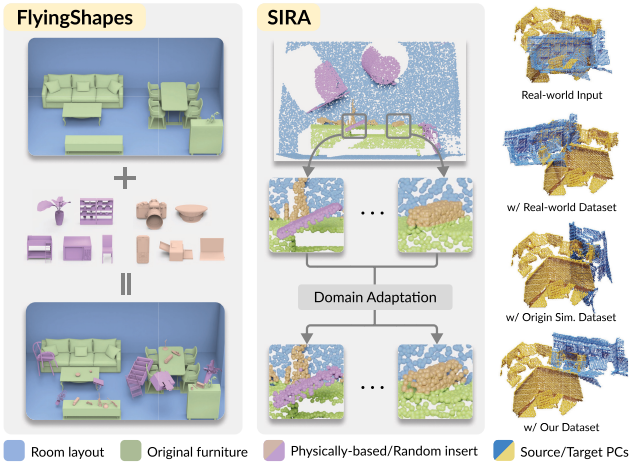

Figure 1. We construct the first large-scale scene-level synthetic dataset for point cloud registration, called FlyingShapes, by inserting objects into simple indoor scenes through physically-based and random strategies. Then, a generative-based pipeline, named SIRA, is designed for sim-to-real adaptation. Qualitative results show the performance improvement of our approach.

1. Introduction

3D point cloud registration is a fundamental task, receiving incredible attention from both the industry and academia, due to its wide applications in robotics [19, 42, 39], graphics [85], computer vision [14, 27], *etc*. Given a partially overlapping point cloud pair, the algorithm is required to predict a 3D rigid transformation to align them.

Recent data-driven deep-learning-based methods have attained remarkable success in indoor scenarios [6, 12, 13, 29, 80, 79]. To support the training, commonly-used datasets can be divided into two categories: (i) object level, *e.g*., ModelNet40 [70] and ShapeNet [10], and (ii) indoor-scene level, *e.g*., 3DMatch [81]. However, the model trained on the synthetic object-level dataset can hardly generalize to real-world indoor scenarios, since the object data has strong shape priors while the geometric structures in indoor scenes are much more complex and diverse. Though directly adopting real-captured datasets can produce satisfying performance, collecting a large amount of data is highly time-consuming. Also, obtaining accurate ground-truth labels is expensive, since the estimated camera poses are prone to errors, so human annotations are needed.

A straightforward avenue to overcome such data scarcity is to utilize simulation data, in which both the data collection and annotation can be done automatically. To leverage existing synthetic 3D indoor scene-level datasets to form the training data for point cloud registration, there are multiple options, *e.g*., SUNCG [60], OpenRooms [40], Structured3D [84], and 3D-FRONT [21]. Specifically, SUNCG has left an apparent void in the community. OpenRooms and Structured3D mainly aim to provide photo-realistic

*Equal contribution.
†Corresponding authors.
[2]Department of Computer Science and Engineering; Institute of Medical Intelligence and XR (IMIXR).

scene images, without providing CAD models. Though 3D-FRONT provides CAD models of rooms and a portion of the furniture, its layouts are relatively simple and more regular than the real ones. To sum up, using existing synthetic data to train a model for point cloud registration has two main challenges: (i) simulated scenes lack complicated geometric structures, since they only contain limited varieties of furniture. A robust feature descriptor and precise feature matching heavily rely on diverse structures in data; and (ii) the point pattern gap, owing to different sensing mechanisms. Simulated scenes tend to contain smooth surfaces with regularly-distributed points, while real scenes exhibit noise and more irregular patterns. Particularly, the feature descriptor is sensitive to the low-level point distribution.

To address challenge (i), inspired by FlowNet [17], we build the first large-scale indoor synthetic dataset named FlyingShapes, on the simulated dataset 3D-FRONT. Specifically, we enrich the local geometric structures by utilizing the object-level ShapeNet dataset as an additional source. We design strategies to randomly arrange objects from ShapeNet on the surfaces of some furniture and in the mid-air of the 3D-FRONT rooms. For challenge (ii), we develop a generative sim-to-real adaptation pipeline, named SIRA. In particular, we design an Adaptive Re-sample Module (ARM) and insert it after different layers of the generator to adaptively adjust point positions in multiple ranges. Lastly, we take the generated point clouds to train the point cloud registration network.

Our main contributions are three-fold:

- We construct the first large-scale indoor synthetic dataset, named FlyingShapes, on the simulated dataset 3D-FRONT for 3D point cloud registration.
- We design a generative sim-to-real adaptation pipeline named SIRA, with an adaptive re-sample module formulated to mitigate the low-level point cloud distribution domain gap. To our best knowledge, this is the first work that exploits domain adaptation strategies to enhance the 3D point cloud registration task.
- Extensive qualitative and quantitative comparisons on both indoor and outdoor datasets show the state-of-the-art performance of our method.

2. Related Work

3D Point Cloud Registration aims to estimate 3D rigid transforms to align point clouds. This task is challenging, mainly due to the partiality of the 3D point clouds. According to the technical pipeline, previous approaches can be divided into two categories: (i) direct registration methods and (ii) correspondence-based methods.

The former category can be further classified into two classes. Some methods [4, 30, 72, 73] regress the transformation from extracted global features of the point clouds. The other methods [67, 68, 34, 78, 23] leverage the differentiable weighted SVD to compute the transformation from the extracted correspondences. These methods obtain satisfying results on synthetic object-level datasets but they can hardly generalize to large-scale scenes as described in [29].

The latter category extracts correspondences between two point clouds and then estimates the transformation using traditional or deep-learning-based pose estimators. Earlier traditional approaches [8, 53, 57, 76, 52, 54] rely on handcrafted features to estimate the correspondences, and apply RANSAC [20] to find the final transformation. Nevertheless, they are easily disturbed by partiality or noise, and RANSAC is time-consuming. More recent methods [12, 25, 6, 3, 47, 65, 80, 79, 51, 66, 75, 38] utilize convolution neural networks to construct more powerful descriptors. Some of them [5, 11, 46] propose deep robust estimators or carefully-designed strategies to improve the accuracy and speed of the transformation calculation. Though these methods achieve prominent performance on various indoor scene datasets, they require large-scale human annotations, which can be expensive and error-prone. This motivates us to address the data scarcity issue using simulation data. Our experimental investigation is built upon the GeoTransformer (GeoTrans) [51] due to its high performance.

Unsupervised Domain Adaptation solves the distribution shift that exists between the source and target domains [24, 41, 7, 59]. Extensive works have been proposed to perform the unsupervised domain adaptation (UDA) on 2D vision tasks [31, 35, 58, 18, 16, 82, 62, 63], while in 3D point cloud tasks, UDA is a developing research topic [74, 69, 9, 33]. Qin *et al*. proposed PointDAN [50], which applies the Maximum Classifier Discrepancy (MCD) [55] to achieve the alignment in feature space, and is the first work to address the point cloud task in the UDA context. Some subsequent works utilized self-supervised learning to solve the domain alignment among representative 3D point cloud tasks, including segmentation [15, 1], detection [77] and classification [86, 2]. For 3D point cloud registration, Horache *et al*. [28] proposed UDGE to generate pairs without ground-truth transformations for unsupervised transfer learning. There are some methods that apply the generative adversarial network (GAN) architecture [26, 37, 36] to perform domain adversarial training and enforce the features of point clouds from both domains to be indistinguishable by the discriminators [56, 83, 71, 45]. We also design a generative-based pipeline to create more realistic point clouds from synthetic data.

3. Method

3.1. Overview

Fig. 2 illustrates the framework of our method. We construct the first large-scale indoor synthetic dataset, Flying-

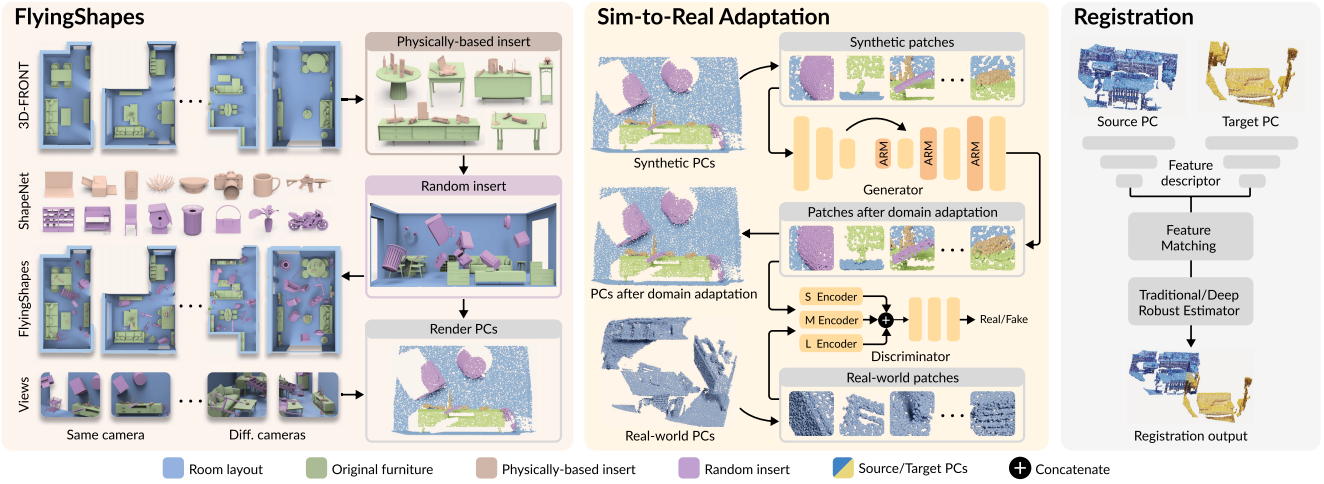

Figure 2. Our SIRA-PCR framework: (i) FlyingShapes is a large-scale indoor synthetic dataset for point cloud registration, which is constructed by applying physically-based and random strategies to insert ShapeNet models into the 3D-FRONT scenes; (ii) SIRA is then applied for sim-to-real domain adaptation on point cloud patches; (iii) the generated synthetic dataset is used to train registration models.

Shapes, for 3D point cloud registration (Sec. 3.2). Then, to further mitigate the domain shifts between FlyingShapes and real-world datasets, rendered point clouds are fed into our SIRA pipeline, where a generative-based sim-to-real domain adaptation network is applied to alleviate the pattern gap (Sec. 3.3). Concurrently, the processed point clouds are used to train the registration methods (Sec. 3.4.). Finally, details of the loss functions are presented in Sec. 3.5.

3.2. FlyingShapes Dataset

To prepare data for training registration methods from synthetic datasets, directly rendering point clouds is unreliable mainly due to two problems: (i) scenes with simple structures have little contributions to improve registration performance; and (ii) low-quality viewpoints result in meaningless point clouds. We carefully design strategies to overcome these issues.

Geometric Augmentation. Although the furniture and room layout in 3D-FRONT is carefully designed by professional designers, compared to real-world layouts, it is still much simpler since the furniture is always placed regularly and only covers a limited portion of categories. To tackle these problems, inspired by FlowNet [17] and FlyingThings3D [44], we augment geometric structure diversity by randomly adding and arranging objects from ShapeNet into scenes of 3D-FRONT. Specifically, the objects are added to scenes in two manners: (i) physically-based insertion. To simulate the cluttered item combinations placed on furniture with a platform, *e.g.*, desks and cabinets, in the real world, we randomly choose and arrange objects on the surfaces of the furniture until collision happens; (ii) random insertion. We randomly scale, rotate, and place objects in the empty space of each room with a specific density of 0.2 objects per cubic meter. In addition, we find that the fraction of different types of geometric structures in the dataset can influence the performance of models. In most cases, planes, which are provided mainly by floors, walls, and ceilings, occupy a much larger fraction of all scenes than other structures; however, they provide limited contributions to the generalization of the feature descriptor and make the network overly focused on simple structures. To this end, we remove flattened planes with a probability of 50% for balancing different structures.

High-quality Viewpoints Selection. Given the scene mesh, finding a proper viewpoint to capture an informative point cloud automatically is non-trivial. Although 3D-FRONT provides example camera locations, they contain low-quality views. We first set the depth clip to filter invalid views, which are too close to surfaces. Then, for each viewpoint, the number of materials is utilized to filter the views lacking geometric structures due to few objects. Specifically, we render the instance segmentation map according to different materials, which is further used to count the number of objects within the current view. The view with less than 5 valid objects/furniture is labeled as "invalid" and not taken into our consideration. In addition, since most data in 3DMatch is captured by the human-held camera, we set the height of the cameras to 1.6m to simulate the human visual experience. Also, the range of view is set to [0°, 360°]/[0°, 45°] in the horizontal/vertical direction, and we uniformly sample views with an interval of 30°/15°.

Registration Data Preparation. In order to simulate the occlusion among objects and the imaging procedure of RGB-D cameras, we first adopt a virtual perspective camera to render a depth map for each viewpoint and then convert it into the point cloud. After that, ground-truth correspon-

dences of overlapping point cloud pairs are easily established given accurate camera poses. Following 3DMatch, point cloud pairs whose overlap ratio is below 30% are not taken into consideration.

3.3. Sim-to-Real Adaptation

Given a synthetic point cloud $\mathbf{P}_s \in \mathbb{R}^{N_s \times 3}$ and a real point cloud $\mathbf{P}_r \in \mathbb{R}^{N_r \times 3}$, SIRA is designed to convert $\mathbf{P}_s$ to $\widehat{\mathbf{P}}_s \in \mathbb{R}^{N_s \times 3}$, which has similar point pattern distribution as $\mathbf{P}_r$ and keep the same shape as $\mathbf{P}_s$.

Generator. The generator $\mathcal{G}(\cdot)$ can be seen as an auto-encoder structure, which consists of two parts, *i.e.*, an encoder for extracting features from point clouds and a decoder for recovering points from features. We utilize the KPConv-FPN backbone [61] to extract multi-level features for the point clouds. At the l-th stage of downsampling, downsampled points $\mathbf{P}_s^l \in \mathbb{R}^{N_l \times 3}$ are regarded as superpoints, and their features $\mathbf{F}_s^l \in \mathbb{R}^{N_l \times d_l}$ are extracted from corresponding local patches. As for the $(L-l+1)$-th stage of upsampling, features of superpoints $\mathbf{H}_s^{L-l+1}$ are first assigned to their nearest points in the upsampled dense point cloud $\mathbf{P}_s^{l-1}$. Then, the features $\mathbf{F}_s^l$ are fetched through a skip connection and concatenated with $\mathbf{H}_s^{L-l+1}$. The entire process can be formulated as

$$\widehat{\mathbf{H}}_s^{L-l+1} = \text{Assign}\{\mathbf{H}_s^{L-l+1}, \text{NN}\{\mathbf{P}_s^l, \mathbf{P}_s^{l-1}\}\}, \quad (1)$$

$$\mathbf{H}_s^{L-l} = \text{MLP}(\text{Cat}\{\mathbf{Y}_s^l, \mathbf{F}_s^l, \widehat{\mathbf{H}}_s^{L-l+1}\}), \quad (2)$$

where $\text{Assign}\{.,.\}$ denotes the feature assignment process according to the relationship of the nearest neighbors $\text{NN}\{.,.\}$ between $\mathbf{P}_s^l$ and $\mathbf{P}_s^{l-1}$. $\mathbf{Y}_s^l$ is the output of the l-th ARM, where point positions are adjusted adaptively. $\text{Cat}\{.,.\}$ means the concatenate operation along the feature channel dimension. L is the total number of downsampling/upsampling layers. After that, we use a three-layer MLP as a decoder to regress point cloud coordinates after sim-to-real adaptation.

Adaptive Re-sample Module. Since central point coordinates are subtracted from the local patches, features extracted by KPConv are translation invariant, which means the absolute location information of each point is lost. Consequently, the subsequent decoder cannot recover point clouds with the same shape as the input. A straightforward solution is to concatenate the coordinate of each point to its corresponding feature as positional embedding during upsampling. However, such a practice would limit the variation of the point positions recovered by the decoder and further limit the ability to imitate the point pattern gap. In order to enhance the capability of the generator to adjust point positions, we propose an adaptive re-sample module, ARM, to modify point positions while keeping the outline of a 3D surface made up of points. Several ARMs are inserted into different levels of the generator. For the l-th ARM, given

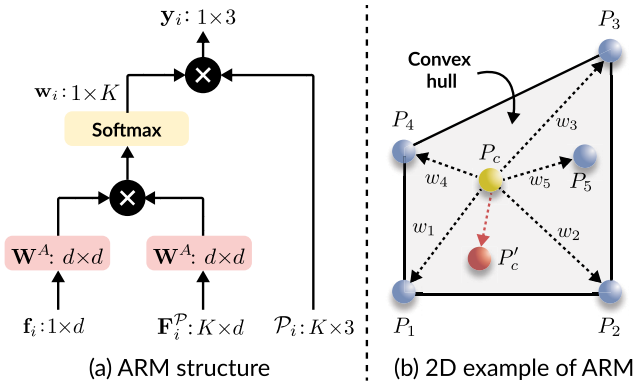

Figure 3. A 2D example and details of our proposed ARM, which re-samples a point P_c to P_c' from the convex hull by adjusting the weights w_i, $i = 1..5$.

the input coordinate matrix $\mathbf{P}_s \in \mathbb{R}^{N_l \times 3}$, we construct a local patch $\mathcal{P}_i$ using the K neighbors for each point $\mathbf{p}_i$. Here, we ignore the superscript l to simplify the notation. Each point of the output coordinate matrix $\mathbf{Y}_s \in \mathbb{R}^{N_l \times 3}$ is the weighted sum of all points within the patch, *i.e.*,

$$\mathbf{y}_i = \sum_{k=1}^{K} w_{i,k} \mathbf{p}_{i,k}^{\mathcal{P}}, \qquad \sum_{k=1}^{K} w_{i,k} = 1, \quad (3)$$

where $\mathbf{p}_{i,k}^{\mathcal{P}} \in \mathcal{P}_i$ is the k-th neighbor point of $\mathbf{p}_i$. $\mathbf{y}_i$ is the re-sampled point. $w_{i,k}$ is the k-th element of $\mathbf{w}_i$ which is computed by the corresponding features of $\mathbf{p}_i$ and $\mathcal{P}_i$,

$$\mathbf{w}_i = \text{softmax}\left(\frac{(\mathbf{F}_i^{\mathcal{P}}\mathbf{W}^A)(\mathbf{f}_i\mathbf{W}^A)^T}{\sqrt{d}}\right). \quad (4)$$

Here, $\mathbf{f}_i \in \mathbb{R}^d$ and $\mathbf{F}_i^{\mathcal{P}} \in \mathbb{R}^{K \times d}$ denote the corresponding features of $\mathbf{p}_i$ and $\mathcal{P}_i$, respectively. $\mathbf{W}^A \in \mathbb{R}^{d \times d}$ is an affine matrix. The detail process is shown in Fig. 3(a). Note that $\mathbf{y}_i$ is the convex combination of $\mathcal{P}_i$, which means its position is within the convex hull of the patch. Fig. 3(b) provides a 2D example for better understanding. The convex hull can well approximate the shape of its corresponding local patch $\mathcal{P}_i$ if K is chosen small enough, e.g., 10, and $\mathbf{y}_i$ can be further regarded as a re-sampled point in that patch.

Multi-scale Discriminator. Though PointNet [49] is the most suitable structure of discriminator as is analyzed in previous work [64], directly applying it on the entire point cloud is not appropriate in our setting. Since SIRA is applied on small point cloud patches, it requires the discriminator to possess the ability to determine the real/fake classification of the low-level point distribution. However, the extracted global feature from the entire point cloud lacks information on detailed structures. To this end, we first split point clouds into local patches, whose respective field is similar to SIRA. A patch-level discriminator $\mathcal{D}(\cdot)$ with PointNet architecture is then adopted to discriminate each

patch independently. As the patch is small enough, it enables the network only focuses on the low-level difference between synthetic and real domains without considering the potentially existing semantic information. Moreover, observing that local density may change after re-sampling the point cloud positions, we apply a multi-scale structure to the discriminator to enhance its robustness against different densities. In detail, features of different ranges of patches, *i.e.*, small, medium, and large patches with 5, 10, and 20 points respectively, are extracted and concatenated before being fed into the classifier.

3.4. Registration

Given a source point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$ and a target point cloud $\mathbf{Q} \in \mathbb{R}^{M\times 3}$, the objective of point cloud registration is to attain the rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$ that align the source to the target.

We validate the effectiveness of our approach based on the previous state-of-the-art GeoTrans [51]. It adopts the KPConv-FPN backbone to simultaneously down-sample the input point cloud and extract hierarchical point-wise features. A point-to-node strategy is then used to assign the dense points to coarse nodes, splitting the point cloud into several non-overlapping patches. Further, a coarse-to-fine strategy is used to find overlapping patches between the input point clouds and concurrently to find correspondences between two overlapping patches. Finally, to recover the alignment transformation between the inputs, it designs a local-to-global registration strategy (LGR) to replace RANSAC, which suffers from slow convergence owing to its high iterations. LGR first generates the transformation candidates using the correspondences within each local patch pair. Then, it selects the best transformation by validating it on all correspondences in the entire point cloud, which achieves fast and accurate registration.

3.5. Loss Functions

We adopt different loss functions for the sim-to-real domain adaptation and 3D point cloud registration, respectively. For the former, unlike related works [15, 71] in 3D point cloud semantic segmentation, we have to strictly guarantee the geometric consistency between point clouds before and after modifying by SIRA, since the input point clouds are required to be in the same shape in their overlap regions for registration; otherwise, correspondences become mismatching, which is unreasonable. To achieve this, we introduce Chamfer Distance to constrain the geometric similarity between $\mathbf{P}_s$ and $\widehat{\mathbf{P}}_s$, *i.e.*,

$$\mathcal{L}_{CD} = \sum_{\mathbf{p}_s \in \mathbf{P}_s} \min_{\widehat{\mathbf{p}}_s \in \widehat{\mathbf{P}}_s} \|\mathbf{p}_s - \widehat{\mathbf{p}}_s\|_2^2 + \sum_{\widehat{\mathbf{p}}_s \in \widehat{\mathbf{P}}_s} \min_{\mathbf{p}_s \in \mathbf{P}_s} \|\mathbf{p}_s - \widehat{\mathbf{p}}_s\|_2^2. \quad (5)$$

Besides, we adopt the loss functions in LS-GAN [43], which consists of $\mathcal{L}_G(\cdot)$ and loss $\mathcal{L}_D(\cdot)$ for training the generator and discriminator, respectively. More precisely, $\mathcal{L}_G(\cdot)$ is formulated as

$$\mathcal{L}_G = \mathbb{E}_{\mathbf{P}_s \sim \mathbb{P}_s} \left[(\mathcal{D}(\mathcal{G}(\mathbf{P}_s)) - c)^2 \right], \quad (6)$$

where $\mathbb{P}_s$ denotes the distribution of synthetic point clouds. Meanwhile, $\mathcal{L}_D(\cdot)$ is written as

$$\mathcal{L}_D = \mathbb{E}_{\mathbf{P}_r \sim \mathbb{P}_r} \left[(\mathcal{D}(\mathbf{P}_r) - b)^2 \right] + \mathbb{E}_{\mathbf{P}_s \sim \mathbb{P}_s} \left[(\mathcal{D}(\mathcal{G}(\mathbf{P}_s)) - a)^2 \right], \quad (7)$$

where $\mathbb{P}_r$ denotes the distribution of real point clouds. We set $a = 0$ and $b = c = 1$. In total, the loss function for sim-to-real domain adaptation is

$$\mathcal{L}_{DA} = \mathcal{L}_G + \mathcal{L}_D + \lambda_G \mathcal{L}_{CD}, \quad (8)$$

where we introduce a dynamic weight $\lambda_G = \exp(-10\mathcal{L}_G)$ to balance their contributions.

To train the network for registration, we adopt the loss functions used in [51]. The registration loss $\mathcal{L}_R$ consists of an overlap-aware circle loss $\mathcal{L}_C$ and a point matching loss $\mathcal{L}_P$. Specifically, $\mathcal{L}_C = (\mathcal{L}_C^{\mathbf{P}} + \mathcal{L}_C^{\mathbf{Q}})/2$, where

$$\mathcal{L}_C^{\mathbf{P}} = \frac{1}{|\mathcal{A}|} \sum_{\mathcal{P}_i^{\mathbf{P}} \in \mathcal{A}} \log[1 + \sum_{\mathcal{P}_j^{\mathbf{Q}} \in \varepsilon_p^i} e^{\lambda_i^j \beta_p^{i,j} (d_i^j - \Delta_p)} \cdot \sum_{\mathcal{P}_k^{\mathbf{Q}} \in \varepsilon_n^i} e^{\beta_p^{i,k} (\Delta_n - d_i^k)}]. \quad (9)$$

Here, $\mathcal{A}$ is the set of anchor patches, which is comprised of the patches in $\mathbf{P}$, having at least one positive patch in $\mathbf{Q}$. For each anchor patch $\mathcal{P}_i^{\mathbf{P}} \in \mathcal{A}$, ε_p^i and ε_n^i denotes the set of its positive and negative patches in $\mathbf{Q}$ separately. d_i^j is the distance in the feature space, and $\lambda_i^j = (o_i^j)^{\frac{1}{2}}$ and o_i^j represents the overlap ratio between $\mathcal{P}_i^{\mathbf{P}}$ and $\mathcal{P}_j^{\mathbf{Q}}$. $\beta_p^{i,j} = \gamma(d_i^j - \Delta_p)$ and $\beta_n^{i,k} = \gamma(\Delta_n - d_i^k)$ are the positive and negative weights, respectively. The margins Δ_p and Δ_n are set to 0.1 and 0.4, respectively. The same goes for the loss $\mathcal{L}_C^{\mathbf{Q}}$ on $\mathbf{Q}$.

Moreover, for the i-th single patch, the point matching loss is computed as

$$\mathcal{L}_{P,i} = -\sum_{(u,v)\in\mathcal{M}_i} \log \bar{z}_{u,v}^i - \sum_{u\in\mathcal{I}_i} \log \bar{z}_{u,m_i+1}^i - \sum_{v\in\mathcal{J}_i} \log \bar{z}_{n_i+1,v}^i, \quad (10)$$

where $\mathcal{M}_i$ denotes the set of ground-truth point correspondences extracted with a matching radius τ from each $\hat{\mathcal{C}}_i^*$ and $\{\hat{\mathcal{C}}_i^*\}$ denotes the randomly sampled N_g ground-truth point correspondences. $\bar{z}_{u,v}^i$ is the element in the u-th row and the v-th column of the soft assignment matrix $\bar{\mathbf{Z}}_i \in \mathbb{R}^{(m_i+1)\times(n_i+1)}$. Additionally, $\mathcal{I}_i$ and $\mathcal{J}_i$ denote the sets of unmatched points in the two patches.

4. Experiments

4.1. Experimental Settings

Datasets. We adopt three datasets to build our FlyingShapes dataset. The first one is 3D-FRONT [21], a large-scale dataset of synthetic 3D indoor scenarios, which contains 18,968 rooms with 13,151 CAD 3D furniture objects

Model	3DMatch			3DLoMatch		
	RRE ↓	RTE ↓	RR ↑	RRE ↓	RTE ↓	RR ↑
PerfectMatch [25]	2.199	0.071	78.4	3.528	0.103	33.0
FCGF [12]	1.949	0.066	85.1	3.147	0.100	40.1
D3Feat [6]	2.161	0.067	81.6	3.361	0.103	37.2
PREDATOR [29]	2.029	0.064	89.0	3.048	0.093	59.8
CoFiNet [80]	2.011	0.062	89.3	3.280	0.094	67.5
GeoTrans [51]	1.625	0.053	91.5	2.547	0.074	74.0
RegTR [79]	1.567	**0.049**	92.0	2.827	0.077	64.8
Our SIRA-PCR†	1.609	0.052	91.5	2.474	0.074	67.4
Our SIRA-PCR‡	**1.539**	0.051	**94.1**	**2.388**	**0.072**	**76.6**

Table 1. Registration results on 3DMatch and 3DLoMatch. †: the model is only trained on the synthetic dataset. ‡: the model is fine-tuned on 3DMatch. The best and second-best results are marked in **bold** and underlined for better comparison.

from 3D-FUTURE [22]. The layouts of rooms are created by professional designers and distinctively span 31 scene categories and 34 object semantic super-classes. To enrich the geometric structure, two other datasets are applied as complementary components. ShapeNet [10] is a richly-annotated large-scale repository of object-level 3D CAD models. It contains 55 common object categories, which include about 51,300 unique 3D models. Structured3D [84] is a photo-realistic dataset. It contains 3,500 house designs created by professional designers. We use its depth images to generate point cloud pairs for registration.

To evaluate the effectiveness of different methods, we employ two benchmarks in our experiments. The first one is 3DMatch [81], a widely-used indoor dataset containing 62 scenes among which 46/8/8 scenes are used for training/validation/test, respectively. Following [29], we split its test set to 3DMatch and 3DLoMatch whose point cloud pairs have 30% and 10%-30% overlap. Second is ETH [48], an outdoor dataset containing 4 scenes, which is collected by a laser scanner and includes 713 pairs made up of 132 point clouds. It is extremely challenging due to the luxuriant trees and small facilities in the scenes.

Evaluation metrics. Following [6, 29, 51, 79, 66], we use several metrics to evaluate our method: (i) Registration Recall (RR), the fraction of successfully registered point cloud pairs whose transformation error RMSE is below 0.2m/0.5m for 3D(Lo)Match/ETH; (ii) Inlier Ratio (IR), the fraction of inlier correspondences whose residuals are below 0.1m/0.2m for 3D(Lo)Match/ETH among all hypothesized correspondences; (iii) Feature Matching Recall (FMR), the fraction of point cloud pairs whose IR is higher than 5%; (iv) the median of the average Relative Rotation Error (RRE) and (v) the median of the average Relative Translation Error (RTE) for the successfully registered pairs whose RMSE is below 0.2m/0.5m for 3D(Lo)Match/ETH.

Implementation details. To construct FlyingShapes, we randomly select 200 scenes from 3D-FRONT, generating 107,641 pairs with 21,550 fragments. All rendering processes are operated in Blender. The horizontal/vertical FoV of virtual cameras is set to 60.0°/46.8°. Depth is clipped to [0.3m, 3.0m] and resolution is 640×480. For training SIRA, we choose a similar number of point clouds as the training set of 3DMatch from 25 scenes of FlyingShapes. Note that synthetic point clouds are applied Truncated Signed Distance Function (TSDF) fusion to smooth the density before being fed into SIRA. Adam optimizer [32] is applied to train the network with a batch size of one. All our experiments are built upon GeoTrans [51]. We pre-train it on FlyingShapes with the learning rate set to $1e^{-4}$ for 40 epochs and fine-tune it on the data processed by SIRA with the learning rate set to $5e^{-6}$ for another 20 epochs. We further finetune it on the training set of 3DMatch to obtain SIRA-

# Samples	3DMatch					3DLoMatch				
	5000	2500	1000	500	250	5000	2500	1000	500	250
Feature Matching Recall (%) ↑										
PerfectMatch [25]	95.0	94.3	92.9	90.1	82.9	63.6	61.7	53.6	45.2	34.2
FCGF [12]	97.4	97.3	97.0	96.7	96.6	76.6	75.4	74.2	71.7	67.3
D3Feat [6]	95.6	95.4	94.5	94.1	93.1	67.3	66.7	67.0	66.7	66.5
SpinNet [3]	97.6	97.2	96.8	95.5	94.3	75.3	74.9	72.5	70.0	63.6
PREDATOR [29]	96.6	96.6	96.5	96.3	96.5	78.6	77.4	76.3	75.7	75.3
YOHO [65]	**98.2**	97.6	97.5	97.7	96.0	79.4	78.1	76.3	73.8	69.1
CoFiNet [80]	98.1	98.3	98.1	98.2	98.3	83.1	83.5	83.3	83.1	82.6
GeoTrans [51]	97.9	97.9	97.9	97.9	97.6	88.3	88.6	88.8	**88.6**	**88.3**
OIF-PCR [75]	98.1	98.1	97.9	98.4	98.4	84.6	85.2	85.5	86.6	87.0
RoReg [66]	**98.2**	97.9	98.2	97.8	97.2	82.1	82.1	81.7	81.6	80.2
Our SIRA-PCR†	97.7	97.9	97.7	97.7	97.6	81.7	82.0	82.6	82.6	82.1
Our SIRA-PCR‡	**98.2**	**98.4**	**98.4**	**98.5**	**98.5**	**88.8**	**89.0**	**88.9**	**88.6**	87.7
Inlier Ratio (%) ↑										
PerfectMatch [25]	36.0	32.5	26.4	21.5	16.4	11.4	10.1	8.0	6.4	4.8
FCGF [12]	56.8	54.1	48.7	42.5	34.1	21.4	20.0	17.2	14.8	11.6
D3Feat [6]	39.0	38.8	40.4	41.5	41.8	13.2	13.1	14.0	14.6	15.0
SpinNet [3]	47.5	44.7	39.4	33.9	27.6	20.5	19.0	16.3	13.8	11.1
PREDATOR [29]	58.0	58.4	57.1	54.1	49.3	26.7	28.1	28.3	27.5	25.8
YOHO [65]	64.4	60.7	55.7	46.4	41.2	25.9	23.3	22.6	18.2	15.0
CoFiNet [80]	49.8	51.2	51.9	52.2	52.2	24.4	25.9	26.7	26.8	26.9
GeoTrans [51]	71.9	75.2	76.0	82.2	85.1	**43.5**	45.3	46.2	52.9	57.7
OIF-PCR [75]	62.3	65.2	66.8	67.1	67.5	27.5	30.0	31.2	32.6	33.1
RoReg [66]	**81.6**	**80.2**	75.1	74.1	75.2	39.6	39.6	34.0	31.9	34.5
Our SIRA-PCR†	66.3	74.1	79.8	82.1	83.6	35.9	41.4	47.7	50.4	52.3
Our SIRA-PCR‡	70.8	78.3	**83.7**	**85.9**	**87.4**	43.3	**49.0**	**55.9**	**58.8**	**60.7**
Registration Recall (%) ↑										
PerfectMatch [25]	78.4	76.2	71.4	67.6	50.8	33.0	29.0	23.3	17.0	11.0
FCGF [12]	85.1	84.7	83.3	81.6	71.4	40.1	41.7	38.2	35.4	26.8
D3Feat [6]	81.6	84.5	83.4	82.4	77.9	37.2	42.7	46.9	43.8	39.1
SpinNet [3]	88.6	86.6	85.5	83.5	70.2	59.8	54.9	48.3	39.8	26.8
PREDATOR [29]	89.0	89.9	90.6	88.5	86.6	59.8	61.2	62.4	60.8	58.1
YOHO [65]	90.8	90.3	89.1	88.6	84.5	65.2	65.5	63.2	56.5	48.0
CoFiNet [80]	89.3	88.9	88.4	87.4	87.0	67.5	66.2	64.2	63.1	61.0
GeoTrans [51]	92.0	91.8	91.8	91.4	91.2	75.0	74.8	74.2	74.1	73.5
OIF-PCR [75]	92.4	91.9	91.8	92.1	91.2	**76.1**	**75.4**	**75.1**	**74.4**	**73.6**
RoReg [66]	92.9	93.2	92.7	**93.3**	91.2	70.3	71.2	69.5	67.9	64.3
Our SIRA-PCR†	90.4	89.6	90.6	90.6	90.1	65.0	65.0	64.2	63.3	63.3
Our SIRA-PCR‡	**93.6**	**93.9**	**93.9**	92.7	**92.4**	73.5	73.9	73.0	73.4	71.1

Table 2. Evaluation results on 3DMatch and 3DLoMatch. †: the model is only trained on the synthetic dataset. ‡: the model is fine-tuned on 3DMatch.

# Samples	ETH														
	Feature Matching Recall (%) ↑					Inlier Ratio (%) ↑					Registration Recall (%) ↑				
	5000	2500	1000	500	250	5000	2500	1000	500	250	5000	2500	1000	500	250
PerfectMatch [25]	95.6	94.3	80.5	69.1	51.4	19.7	16.7	12.4	9.3	6.6	81.4	73.5	59.3	46.5	35.0
D3Feat [6]	63.3	71.0	69.7	67.9	60.5	12.5	13.2	13.6	13.5	11.8	59.1	50.4	49.7	44.6	29.1
FCGF [12]	41.1	38.4	32.3	24.6	15.9	5.8	5.3	4.4	3.5	2.8	42.1	36.1	29.5	26.3	18.9
SpinNet [3]	**99.4**	**99.1**	94.6	87.2	66.6	23.2	20.4	15.7	11.9	8.6	96.0	91.1	81.9	71.5	54.3
PREDATOR [29]	65.6	64.5	59.6	52.0	40.5	11.1	10.3	8.5	6.8	5.1	74.7	72.9	67.7	60.3	51.7
CoFiNet [80]	82.5	83.7	81.9	81.1	79.9	9.6	9.8	9.9	9.9	9.8	83.9	82.7	81.9	77.4	68.8
RoReg [66]	96.5	95.6	93.1	92.0	84.1	**28.4**	**25.4**	**21.7**	20.8	17.5	97.1	97.1	95.7	92.3	84.3
GeoTrans [51]	60.5	67.5	75.7	79.0	79.8	6.9	9.2	12.1	13.7	14.6	80.9	76.8	73.3	72.8	70.8
Our SIRA-PCR† + RANSAC	86.4	94.7	**98.1**	**99.0**	**98.7**	9.9	13.8	18.7	**21.7**	**24.1**	96.4	95.9	97.3	96.2	95.4
Our SIRA-PCR‡ + RANSAC	79.2	87.3	93.6	94.7	93.2	8.7	11.8	15.7	17.6	18.5	93.8	91.8	88.4	85.1	82.2
Our SIRA-PCR† + LGR	—		85.1	—		—		9.3	—		—		**99.0**	—	
Our SIRA-PCR‡ + LGR	—		79.0	—		—		8.5	—		—		97.1	—	

Table 3. Evaluation results on ETH. †: the model is only trained on the synthetic dataset. ‡: the model is fine-tuned on 3DMatch. —: results with different samples are not applicable for LGR since it uses all correspondences.

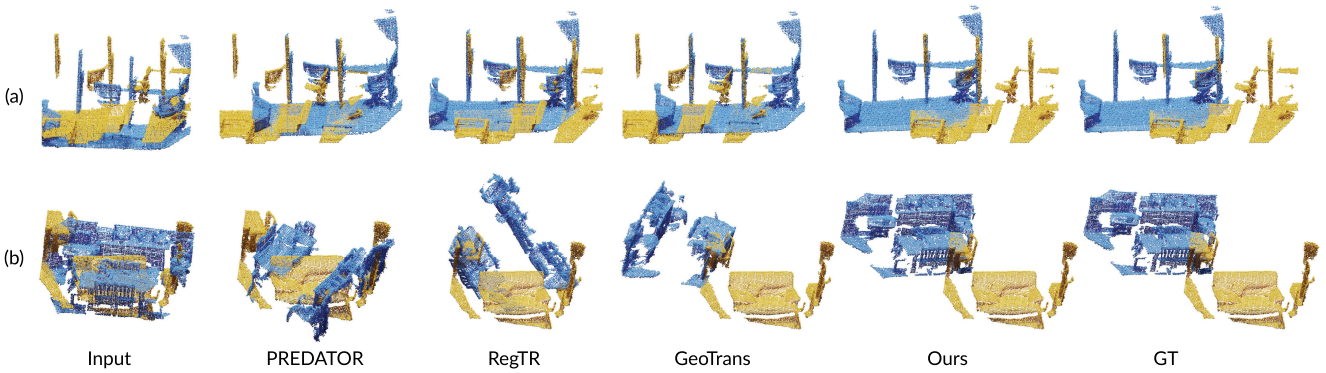

Figure 4. Qualitative comparison with state-of-the-art methods [29, 79, 51] on (a) 3DMatch and (b) 3DLoMatch.

PCR‡. All experiments run on four NVIDIA Tesla P40s. We set the neighbor numbers to 100/30/30/30 for the corresponding four stages of the KPConv backbone. Other hyperparameters are the same as [51]. For the results obtained by RANSAC, we follow [6, 29, 51, 66], running 50k iterations to estimate the transformation, and the inlier distance threshold is set to 0.05m/0.4m for 3D(Lo)Match/ETH.

4.2. Evaluation on Indoor Benchmark

We first compare the registration results of our method with the recent state of the arts in Tab. 1. Following [6, 12, 29, 51, 79], we report RRE and RTE for successfully registered point-cloud pairs. Without training on any real-world dataset, our SIRA-PCR achieves even better performances on the 3DMatch test set *i.e.*, lower RRE and RTE under the same RR, compared to our baseline GeoTrans [51]. Similar to [71, 15], we further fine-tune our model on the 3DMatch training set. It shows that our model clearly outperforms our competitors. In particular, it improves the RR by 2.1/2.6 percentage points (pp) on 3DMatch/3DLoMatch compared to the second-best method and achieves lower RRE and RTE, which demonstrates the effectiveness of our approach.

We also evaluate the correspondence results compared to other methods [25, 12, 6, 3, 29, 65, 80, 51, 75, 66] under different numbers of sampled correspondences with RANSAC, as shown in Tab. 2. It shows that our method surpasses almost all competitors on the FMR and IR metrics. The performance of RR drops on 3DLoMatch since RANSAC is more sensitive to the inlier distribution than LGR. With a high inlier ratio, RANSAC and LGR show similar performance; however, the performance obtained by RANSAC degrades due to the uneven distribution of inliers, where correspondences may concentrate on some local patches. Fig. 4 provides the qualitative comparison. Our method performs better in challenging cases, such as repeated geometric structures and low overlap ratio, which demonstrates its robustness.

4.3. Evaluation on Outdoor Benchmark

To evaluate the generalization, we conduct experiments on the ETH dataset, as shown in Tab. 3. Following [66], we directly use the models trained on 3DMatch without fine-tuning. We also release the performance of our method only trained on our FlyingShapes dataset. Without considering

Model	3DMatch			3DLoMatch		
	RRE ↓	RTE ↓	RR ↑	RRE ↓	RTE ↓	RR ↑
(a) Structured3D	1.695	0.055	85.5	2.852	0.078	48.6
(b) 3DFront	1.777	0.052	74.8	2.990	0.083	33.5
(c) + ShapeNet	**1.660**	**0.049**	85.6	2.672	0.076	55.1
(d) + Delete planes	1.675	0.052	87.0	2.638	**0.074**	57.4
(e) + Structured3D	1.672	0.051	**87.5**	**2.521**	**0.074**	**63.8**

Table 4. Ablation studies of FlyingShapes.

Model	3DMatch			3DLoMatch		
	RRE ↓	RTE ↓	RR ↑	RRE ↓	RTE ↓	RR ↑
(a) baseline	1.672	**0.051**	87.5	2.521	**0.074**	63.8
(b) uniform noise	**1.589**	**0.051**	88.6	2.570	0.075	62.7
(c) Gaussian noise	1.601	0.052	89.3	2.595	0.080	63.8
(d) TSDF	1.691	**0.051**	87.9	2.539	0.075	63.9
(e) w/o ARM	1.606	0.052	90.8	2.519	0.077	65.3
(f) w/o multi-scale	1.625	0.052	91.2	2.482	**0.074**	66.6
(g) SIRA†	1.609	0.052	**91.5**	**2.474**	**0.074**	**67.4**

Table 5. Ablation studies of SIRA. †: the model is only trained on the synthetic dataset.

sample numbers, our method achieves the highest FMR, IR, and RR. In terms of the RR metric, our method attains the best performance which outperforms the second best by 1.9 pp, showing well generalization ability on outdoor scenes. Interestingly, unlike the results on 3DMatch shown in Tab. 1 and Tab. 2, our models only trained on the synthetic dataset outperform those fine-tuned on 3DMatch, which implies that our FlyingShapes dataset benefits to improve the robustness and generalization ability of models. The qualitative comparison is provided in Fig. 5. Our approach solves the challenging case with unseen complex structures.

4.4. Ablation Studies

Extensive experiments are conducted on 3DMatch to understand the role of each component in our method.

Different Settings of FlyingShapes. We first show how we generate an informative dataset from an unreliable synthetic dataset in Tab. 4. The first two rows show that directly training models on Structured3D and 3D-FRONT lead to poor results. After inserting ShapeNet models using our designed strategies, it brings a large improvement in the RR metric on 3DLoMatch. An additional performance gain is achieved by deleting planes in the scenes since the fraction of different geometric structures becomes more balanced. Finally, Row (e) shows that combining Structured3D enhances the feature description ability.

Necessity of SIRA. SIRA plays an important role in our framework. First, we conduct experiments to demonstrate its necessity. The model pre-trained on FlyingShapes is regarded as the baseline. To find out whether the gap between the real and synthetic domains can be simply eliminated by adding noises, we trained models on FlyingShapes with different types of noises whose magnitudes are close to the

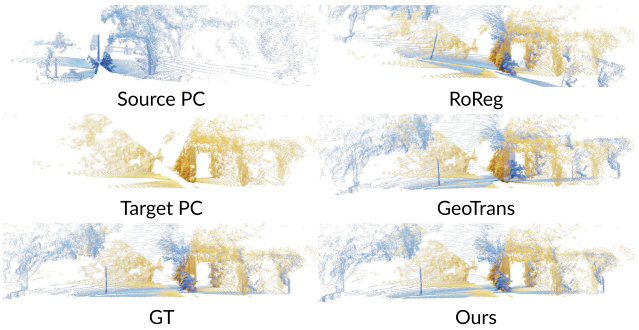

Figure 5. Qualitative comparison with representative state-of-the-art methods [51, 66] on the ETH.

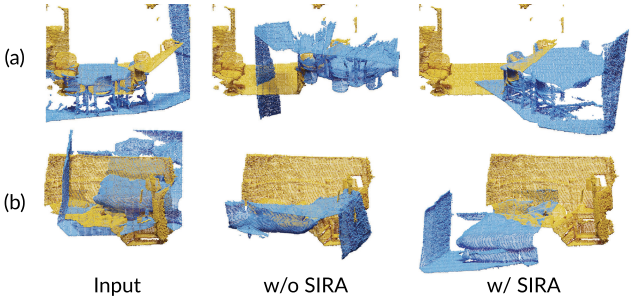

Figure 6. Qualitative comparison between models with/without SIRA on challenging cases in 3DLoMatch.

ones generated by SIRA. As is shown in Tab. 5, comparing Rows (b) and (c), we find that common noises, *e.g.*, uniform noise and Gaussian noise, can only bring a tiny improvement on 3DMatch, which indicates that the pattern gap has not been mitigated. Moreover, comparing Rows (a) and (e) shows that fine-tuning on the data processed by TSDF fusion advances little performance, either. In a word, Rows (a-d) mean that the domain gap can hardly be alleviated by adopting existing common operations, which indicates the necessity of our proposed SIRA. After using SIRA to process the data before fine-tuning, the performance shown in Row (g) outperforms the baseline by a large margin, which demonstrates the domain gap has been mitigated. Fig. 6 illustrates some challenging cases which are confusing for the model due to extremely low overlap and lack of salient geometric structures in the overlap regions. With SIRA adopted, the features extracted by models become strong enough to solve these cases.

Components in SIRA. To better understand our proposed components in SIRA, we ablate each of them separately. First, we remove ARM and use the original point coordinates to concatenate with features in the generator. Comparing Rows (e) and (g), consistent performance decrements are observed on both 3DMatch and 3DLoMatch, which reflects the effectiveness of ARM. Next, Comparing Rows (f) and (g), introducing the multi-scale structure boosts the performance, which means multiple receptive fields provide extra useful information for the discriminator.

5. Conclusion

We present SIRA-PCR, a new 3D point cloud registration framework that enables models trained on synthetic datasets to achieve better performance without relying on any real-world dataset. To the best of our knowledge, we build the first large-scale indoor synthetic dataset, FlyingShapes, for point cloud registration. Besides, we first explore the sim-to-real adaptation, SIRA, to alleviate the low-level domain gap in this task. Experimental results also confirm the state-of-the-art performance and robustness of SIRA-PCR on both indoor and outdoor benchmarks.

Acknowledgments This work was supported by the National Natural Science Foundation of China (NSFC) under grant No.62071097, the Sichuan Science and Technology Program of China under grants Nos.2023NSFSC0458, 2023NSFSC0462 and 2023NSFSC1972, and the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Reference Number T45-401/22-N).

References

[1] Idan Achituve, Haggai Maron, and Gal Chechik. Self-supervised learning for domain adaptation on point clouds. In *Proc. WACV*, pages 123–133, 2021. 2

[2] Antonio Alliegro, Davide Boscaini, and Tatiana Tommasi. Joint supervised and self-supervised learning for 3D real world challenges. In *Proc. ICPR*, pages 6718–6725, 2021. 2

[3] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. SpinNet: Learning a general surface descriptor for 3D point cloud registration. In *Proc. CVPR*, pages 11753–11762, 2021. 2, 6, 7

[4] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. PointNetLK: Robust & efficient point cloud registration using PointNet. In *Proc. CVPR*, pages 7163–7172, 2019. 2

[5] 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 *Proc. CVPR*, pages 15859–15869, 2021. 2

[6] 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 *Proc. CVPR*, pages 6359–6367, 2020. 1, 2, 6, 7

[7] Mahsa Baktashmotlagh, Mehrtash T. Harandi, Brian C. Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proc. ICCV*, pages 769–776, 2013. 2

[8] Paul J. Besl and Neil D. McKay. A method for registration of 3D shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 2

[9] Adriano Cardace, Riccardo Spezialetti, Pierluigi Zama Ramirez, Samuele Salti, and Luigi Di Stefano. Self-distillation for unsupervised 3D domain adaptation. In *Proc. WACV*, pages 4166–4177, 2023. 2

[10] Angel X Chang, Thomas Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 6

[11] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proc. CVPR*, pages 2514–2523, 2020. 2

[12] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proc. ICCV*, pages 8958–8966, 2019. 1, 2, 6, 7

[13] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPFNet: Global context aware local features for robust 3D point matching. In *Proc. CVPR*, pages 195–205, 2018. 1

[14] Jean-Emmanuel Deschaud. IMLS-SLAM: Scan-to-model matching based on 3D data. In *Proc. ICRA*, pages 2480–2485. IEEE, 2018. 1

[15] Runyu Ding, Jihan Yang, Li Jiang, and Xiaojuan Qi. DODA: Data-oriented sim-to-real domain adaptation for 3D semantic segmentation. In *Proc. ECCV*, pages 284–303. Springer, 2022. 2, 5, 7

[16] Jiahua Dong, Yang Cong, Gan Sun, Bineng Zhong, and Xiaowei Xu. What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation. In *Proc. CVPR*, pages 4023–4032, 2020. 2

[17] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proc. ICCV*, pages 2758–2766, 2015. 2, 3

[18] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *Proc. CVPR*, pages 3937–3946, 2021. 2

[19] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3D point cloud registration for localization using a deep neural network auto-encoder. In *Proc. CVPR*, pages 4631–4640, 2017. 1

[20] 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*, 24(6):381–395, 1981. 2

[21] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3D-FRONT: 3D furnished rooms with layouts and semantics. In *Proc. ICCV*, pages 10933–10942, 2021. 1, 5

[22] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3D-FUTURE: 3D furniture shape with texture. *International Journal of Computer Vision*, pages 1–25, 2021. 6

[23] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust point cloud registration framework based on deep graph matching. In *Proc. CVPR*, pages 8893–8902, 2021. 2

[24] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proc. ICML*, pages 1180–1189, 2015. 2

[25] Zan Gojcic, Caifa Zhou, Jan D. Wegner, and Andreas Wieser. The perfect match: 3D point cloud matching with smoothed densities. In *Proc. CVPR*, pages 5545–5554, 2019. 2, 6, 7

[26] Ian Goodfellow, Jean Pougetabadie, Mehdi Mirza, Bing Xu, David Wardefarley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NeurIPS*, pages 2672–2680, 2014. 2

[27] Lei Han, Lan Xu, Dmytro Bobkov, Eckehard Steinbach, and Lu Fang. Real-time global registration for globally consistent RGB-D slam. *IEEE Trans. on Robotics*, 35(2):498–508, 2019. 1

[28] Sofiane Horache, Jean-Emmanuel Deschaud, and François Goulette. 3D point cloud registration with multi-scale architecture and unsupervised transfer learning. In *2021 International Conference on 3D Vision (3DV)*, pages 1351–1361. IEEE, 2021. 2

[29] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. PREDATOR: Registration of 3D point clouds with low overlap. *arXiv:2011.13005*, 2020. 1, 2, 6, 7

[30] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In *Proc. CVPR*, pages 11366–11374, 2020. 2

[31] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proc. CVPR*, pages 4893–4902, 2019. 2

[32] P. Diederik Kingma and Lei Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015. 6

[33] Lingdong Kong, Niamul Quader, and Venice Erin Liong. Conda: Unsupervised domain adaptation for LiDAR segmentation via regularized domain concatenation. *arXiv preprint arXiv:2111.15242*, 2021. 2

[34] Jiahao Li, Changhao Zhang, Ziyao Xu, Hangning Zhou, and Chi Zhang. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In *Proc. ECCV*, pages 378–394, 2020. 2

[35] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proc. CVPR*, pages 9641–9650, 2020. 2

[36] Ru Li, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. JigsawGAN: Auxiliary learning for solving jigsaw puzzles with generative adversarial networks. *IEEE Trans. on Image Processing*, 31:513–524, 2021. 2

[37] Ru Li, Chi-Hao Wu, Shuaicheng Liu, Jue Wang, Guangfu Wang, Guanghui Liu, and Bing Zeng. SDP-GAN: Saliency detail preservation generative adversarial networks for high perceptual quality style transfer. *IEEE Trans. on Image Processing*, 30:374–385, 2020. 2

[38] Yang Li and Tatsuya Harada. Lepard: Learning partial point cloud matching in rigid and deformable scenes. *Proc. CVPR*, 2022. 2

[39] Ying Li, Lingfei Ma, Zilong Zhong, Fei Liu, Michael A Chapman, Dongpu Cao, and Jonathan Li. Deep learning for LiDAR point clouds in autonomous driving: A review. *IEEE Trans. on Neural Networks and Learning Systems*, 32(8):3412–3432, 2020. 1

[40] Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh Gundavarapu, Jia Shi, et al. OpenRooms: An open framework for photorealistic indoor scene datasets. In *Proc. CVPR*, pages 7190–7199, 2021. 1

[41] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. *Proc. NeurIPS*, 29:136–144, 2016. 2

[42] Weixin Lu, Yao Zhou, Guowei Wan, Shenhua Hou, and Shiyu Song. L3-Net: Towards learning based LiDAR localization for autonomous driving. In *Proc. CVPR*, pages 6389–6398, 2019. 1

[43] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017. 5

[44] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proc. CVPR*, pages 4040–4048, 2016. 3

[45] Kazuto Nakashima, Yumi Iwashita, and Ryo Kurazume. Generative range imaging for learning scene priors of 3D LiDAR data. In *Proc. WACV*, pages 1256–1266, 2023. 2

[46] 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 *Proc. CVPR*, pages 7193–7203, 2020. 2

[47] Fabio Poiesi and Davide Boscaini. Learning general and distinctive 3D local deep descriptors for point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[48] François Pomerleau, Ming Liu, Francis Colas, and Roland Siegwart. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*, 31(14):1705–1711, 2012. 6

[49] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proc. CVPR*, pages 652–660, 2017. 4

[50] Can Qin, Haoxuan You, Lichen Wang, C.-C. Jay Kuo, and Yun Fu. PointDAN: A multi-scale 3D domain adaption network for point cloud representation. *Proc. NeurIPS*, 32, 2019. 2

[51] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11143–11152, June 2022. 2, 5, 6, 7, 8

[52] Szymon Rusinkiewicz. A symmetric objective function for LiDAR. *ACM Trans. on Graphics*, 38(4):1–7, 2019. 2

[53] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 145–152, 2001. 2

[54] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3D registration. In *Proc. ICRA*, pages 3212–3217, 2009. 2

[55] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proc. CVPR*, pages 3723–3732, 2018. 2

[56] Khaled Saleh, Ahmed Abobakr, Mohammed Attia, Julie Iskander, Darius Nahavandi, Mohammed Hossny, and Saeid Nahvandi. Domain adaptation for vehicle detection from bird's eye view LiDAR point cloud data. In *Proc. ICCVW*, pages 0–0, 2019. 2

[57] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-LiDAR. In *Robotics: Science and Systems*, volume 2, page 435, 2009. 2

[58] Kendrick Shen, Robbie M Jones, Ananya Kumar, Sang Michael Xie, Jeff Z HaoChen, Tengyu Ma, and Percy Liang. Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation. In *Proc. ICML*, pages 19847–19878, 2022. 2

[59] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018. 2

[60] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proc. CVPR*, pages 1746–1754, 2017. 1

[61] 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 *Proc. ICCV*, pages 6411–6420, 2019. 4

[62] Marco Toldo, Andrea Maracani, Umberto Michieli, and Pietro Zanuttigh. Unsupervised domain adaptation in semantic segmentation: A review. *Technologies*, 8(2):35, 2020. 2

[63] Alessio Tonioni, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Unsupervised domain adaptation for depth prediction from images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 42(10):2396–2409, 2019. 2

[64] He Wang, Zetian Jiang, Li Yi, Kaichun Mo, Hao Su, and Leonidas J. Guibas. Rethinking sampling in 3D point cloud generative adversarial networks. *arXiv preprint arXiv:2006.07029*, 2020. 4

[65] Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In *ACM Trans. on Multimedia*, pages 1630–1641, 2022. 2, 6, 7

[66] Haiping Wang, Yuan Liu, Qingyong Hu, Bing Wang, Jianguo Chen, Zhen Dong, Yulan Guo, Wenping Wang, and Bisheng Yang. RoReg: Pairwise point cloud registration with oriented descriptors and local rotations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2, 6, 7, 8

[67] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration. In *Proc. ICCV*, pages 3523–3532, 2019. 2

[68] Yue Wang and Justin M. Solomon. PRNet: Self-supervised learning for partial-to-partial registration. In *Proc. NeurIPS*, pages 8814–8826, 2019. 2

[69] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud. In *Proc. ICRA*, pages 4376–4382, 2019. 2

[70] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proc. CVPR*, pages 1912–1920, 2015. 1

[71] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer learning from synthetic to real LiDAR point cloud for semantic segmentation. In *Proc. AAAI*, volume 36, pages 2795–2803, 2022. 2, 5, 7

[72] Hao Xu, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. OMNet: Learning overlapping mask for partial-to-partial point cloud registration. In *Proc. ICCV*, 2021. 2

[73] Hao Xu, Nianjin Ye, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. FINet: Dual branches feature interaction for partial-to-partial point cloud registration. In *Proc. AAAI*, volume 36, pages 2848–2856, 2022. 2

[74] Qiangeng Xu, Yin Zhou, Weiyue Wang, Charles R. Qi, and Dragomir Anguelov. SPG: Unsupervised domain adaptation for 3D object detection via semantic point generation. In *Proc. ICCV*, pages 15446–15456, 2021. 2

[75] Fan Yang, Lin Guo, Zhi Chen, and Wenbing Tao. One-Inlier is First: Towards efficient position encoding for point cloud registration. In *Advances in Neural Information Processing Systems*. 2, 6, 7

[76] Jiaolong Yang, Hongdong Li, and Yunde Jia. Go-ICP: Solving 3D registration efficiently and globally optimally. In *Proc. CVPR*, pages 1457–1464, 2013. 2

[77] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D: Self-training for unsupervised domain adaptation on 3D object detection. In *Proc. CVPR*, pages 10368–10378, 2021. 2

[78] Zi Jian Yew and Gim Hee Lee. RPM-Net: Robust point matching using learned features. In *Proc. CVPR*, pages 11824–11833, 2020. 2

[79] Zi Jian Yew and Gim Hee Lee. RegTr: End-to-end point cloud correspondences with transformers. In *Proc. CVPR*, pages 6677–6686, 2022. 1, 2, 6, 7

[80] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. CoFiNet: Reliable coarse-to-fine correspondences for robust point cloud registration. *Proc. NeurIPS*, 34:23872–23884, 2021. 1, 2, 6, 7

[81] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. In *Proc. CVPR*, pages 1802–1811, 2017. 1, 6

[82] Yifan Zhang, Ying Wei, Qingyao Wu, Peilin Zhao, Shuaicheng Niu, Junzhou Huang, and Mingkui Tan. Collaborative unsupervised domain adaptation for medical image diagnosis. *IEEE Trans. on Image Processing*, 29:7834–7844, 2020. 2

[83] Sicheng Zhao, Yezhen Wang, Bo Li, Bichen Wu, Yang Gao, Pengfei Xu, Trevor Darrell, and Kurt Keutzer. EPointDa: An end-to-end simulation-to-real domain adaptation framework

for LiDAR point cloud segmentation. In *Proc. AAAI*, pages 3500–3509, 2021. 2

[84] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3D: A large photo-realistic dataset for structured 3D modeling. In *Proc. ECCV*, pages 519–535, 2020. 1, 6

[85] Runsong Zhu, Di Kang, Ka-Hei Hui, Yue Qian, Xuefei Zhe, Zhen Dong, Linchao Bao, and Chi-Wing Fu. Semi-signed neural fitting for surface reconstruction from unoriented point clouds. *arXiv preprint arXiv:2206.06715*, 2022. 1

[86] Longkun Zou, Hui Tang, Ke Chen, and Kui Jia. Geometry-aware self-training for unsupervised domain adaptation on object point clouds. In *Proc. ICCV*, pages 6403–6412, 2021. 2