Learning-based Point Cloud Registration for 6D Object Pose Estimation in the Real World

Zheng Dang[1], Lizhou Wang[3], Yu Guo[3], and Mathieu Salzmann[1,2]

CVLab, EPFL, Switzerland
Clearspace, Switzerland
{zheng.dang, mathieu.salzmann}@epfl.ch
Xi'an Jiaotong University, Shaanxi, China
dzyxwanglizhou@stu.xjtu.edu.cn, yu.guo@xjtu.edu.cn

Abstract. In this work, we tackle the task of estimating the 6D pose of an object from point cloud data. While recent learning-based approaches to addressing this task have shown great success on synthetic datasets, we have observed them to fail in the presence of real-world data. We thus analyze the causes of these failures, which we trace back to the difference between the feature distributions of the source and target point clouds, and the sensitivity of the widely-used SVD-based loss function to the range of rotation between the two point clouds. We address the first challenge by introducing a new normalization strategy, Match Normalization, and the second via the use of a loss function based on the negative log likelihood of point correspondences. Our two contributions are general and can be applied to many existing learning-based 3D object registration frameworks, which we illustrate by implementing them in two of them, DCP and IDAM. Our experiments on the real-scene TUD-L [23], LINEMOD [20] and Occluded-LINEMOD [7] datasets evidence the benefits of our strategies. They allow for the first time learning-based 3D object registration methods to achieve meaningful results on real-world data. We therefore expect them to be key to the future development of point cloud registration methods.

Keywords: 6D Object Pose Estimation, Point Cloud Registration

1 Introduction

Estimating the 6D pose, i.e., 3D rotation and 3D translation, of an object has many applications in various domains, such as robotics grasping, simultaneous localization and mapping (SLAM), and augmented reality. In this context, great progress has been made by learning-based methods operating on RGB(D) images [29,44,40,54,38,70,58,33,57,53]. In particular, these methods achieve impressive results on real-world images.

In parallel to this line of research, and thanks to the development of point cloud processing networks [42,43,4,60,32,52,69], several learning-based 3D object registration algorithms [3,61,62,67,68] have emerged to estimate 6D object poses from 3D measurements only, such as those obtained with a LiDAR. Since they

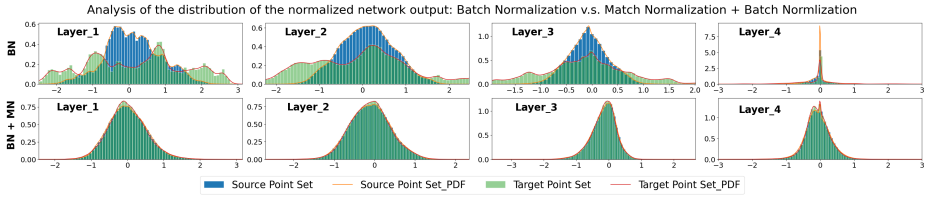

Fig. 1: **Feature distributions at different network layers with real-world data. Top:** With Batch Normalization, the distributions of the features extracted from the model (source) and input (target) point clouds differ significantly. **Bottom:** Our Match Normalization strategy makes these distributions much more similar.

focus solely on 3D information, discarding any RGB appearance, these methods have demonstrated excellent generalization to previously unseen objects. However, in contrast with *scene-level* registration methods [10,11,14,13,24,9], these *object-level* learning-based techniques are typically evaluated only on synthetic datasets, and virtually never on real-world data, such as the TUD-L [23], LineMod [20] and LineMod-Occluded [7] datasets.

In our experiments with the state-of-the-art learning-based object-level registration frameworks, we observed them to struggle with the following challenges. First, in contrast with synthetic datasets where all objects have been normalized to a common scale, the size of different objects in the real world varies widely. The fact that the sensor depicts only an unknown portion the object precludes a simple re-scaling of the target point cloud. In synthetic data, the target point cloud is typically sampled from the normalized model, thus ignoring this difficulty. Second, while synthetic datasets typically limit the rotation between the source and target point clouds in the 45° range, real-world sensors may observe the target object from any viewpoint, covering the full rotation range.

As shown in the top row of Figure 1, the first above-mentioned challenge translates to a significant gap between the feature distributions of the source and target point clouds in the inner layers of the network. The greater the difference between the two distributions, the fewer correct inlier matches will be found, which then yields a decrease in performance. To address this, we propose a new normalization method, which we refer to as Match Normalization. Match Normalization exploits an instance-level scale parameter in each layer of the feature extraction network. This parameter is shared by the source and target point clouds, thus providing robustness to partial observations and outliers. This makes the distributions of the two point clouds more concentrated and similar, as shown in the bottom row of Figure 1, and yields increase the number of inlier matches, as evidenced by our experiments.

Furthermore, we observed the second above-mentioned challenge to cause instabilities in the network convergence when relying on the widely-used SVD-based loss function [61,62,67] for training. To address this, we exploit a simple negative log-likelihood (NLL) loss function, which we show to improve convergence and lead to better object-level pose estimation accuracy.

Altogether, our contributions have the following advantages: (i) The proposed Match Normalization is applicable to many point cloud registration network architectures; (ii) Both contributions only involve small changes to the network and yet substantially improve its performance on real object-level pose estimation datasets; (iii) They allow for the first time a learning-based point cloud registration method to achieve meaningful results on real-world 6D object pose estimation datasets, such as the TUD-L [23], LINEMOD [20] and Occluded-LINEMOD [7] datasets. We will make our code publicly available upon acceptance of the paper.

2 Related Work

Traditional Point Cloud Registration Methods. The Iterative Closest Point (ICP) [5] is the best-known local registration methods.Several variants, such as Generalized-ICP [51] and Sparse ICP [6], have been proposed to improve robustness to noise and mismatches, and we refer the reader to [41,47] for a complete review of ICP-based strategies. The main drawback of these methods is their requirement for a reasonable initialization to converge to a good solution. Only relatively recently has this weakness been addressed by the globally-optimal registration method Go-ICP [66]. In essence, this approach follows a branch-and-bound strategy to search the entire 3D motion space $SE(3)$. A similar strategy is employed by the sampling-based global registration algorithm Super4PCS [36]. While globally optimal, Go-ICP come at a much higher computational cost than vanilla ICP. This was, to some degree, addressed by the Fast Global Registration (FGR) algorithm [71], which leverages a local refinement strategy to speed up computation. While effective, FGR still suffers from the presence of noise and outliers in the point sets, particularly because, as vanilla ICP, it relies on 3D point-to-point distance to establish correspondences. In principle, this can be addressed by designing point descriptors that can be more robustly matched. For example, [56] relies on generating pose hypotheses via feature matching, followed by a RANSAC-inspired method to choose the candidate pose with the largest number of support matches. Similarly, TEASER [64] and its improved version TEASER++ [65] take putative correspondences obtained via feature matching as input and remove the outlier ones by an adaptive voting scheme. In addition to the above, there are many algorithms [2,36,45,37,16,35,46,26,27,49,48,64,30,1,21,19,17] that have contributed to this direction.

Learning-based Object Point Cloud Registration Methods. A key requirement to enable end-to-end learning-based registration was the design of deep networks acting on unstructured sets. Deep sets [69] and PointNet [42] constitute the pioneering works in this direction. In particular, PointNetLK [3] combines the PointNet backbone with the traditional, iterative Lucas-Kanade (LK) algorithm [34] so as to form an end-to-end registration network; DCP [61] exploits DGCNN [60] backbones followed by Transformers [55] to establish 3D-3D correspondences, which are then passed through an SVD layer to obtain the

final rigid transformation. While effective, PointNetLK and DCP cannot tackle the partial-to-partial registration scenario. That is, they assume that both point sets are fully observed, during both training and test time. This was addressed by PRNet [62] via a deep network designed to extract keypoints from each input set and match these keypoints. This network is then applied in an iterative manner, so as to increasingly refine the resulting transformation. IDAM [31] builds on the same idea as PRNet, using a two-stage pipeline and a hybrid point elimination strategy to select keypoints. By contrast, RPM-Net [67] builds on DCP and adopts a different strategy, replacing the softmax layer with an optimal transport one so as to handle outliers. Nevertheless, as PRNet, RPM-Net relies on an iterative strategy to refine the computed transformation. DeepGMR [68] leverages mixtures of Gaussians and formulates registration as the minimization of the KL-divergence between two probability distributions to handle outliers. In any event, the methods discussed above were designed to handle point-clouds in full 3D, and were thus neither demonstrated for registration from 2.5D measurements, nor evaluated on real scene datasets, such as TUD-L, LINEMOD and Occluded-LINEMOD. In this work, we identify and solve the issues that prevent the existing learning-based methods from working on real-world data, and, as a result, develop the first learning-based point cloud registration method able to get reasonable result on the real-world 6D pose estimation datasets.

3 Methodology

3.1 Problem Formulation

Let us now introduce our approach to object-level 3D registration. We consider the problem of partial-to-partial registration between two point clouds $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$ and $\mathcal{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_N\}$, which are two sets of 3D points sampled from the same object surface, with $\mathbf{x}_i, \mathbf{y}_j \in \mathbb{R}^3$. We obtain the source point set $\mathcal{X}$ by uniform sampling from the mesh model, and the target one $\mathcal{Y}$ from a depth map I_{depth} acquired by a depth sensor, assuming known camera intrinsic parameters. We typically refer to $\mathcal{X}$ as the source point set and to $\mathcal{Y}$ as the target point set. Our goal is to estimate the rotation matrix $\mathbf{R} \in SO(3)$ and translation vector $\mathbf{t} \in \mathbb{R}^3$ that align $\mathcal{X}$ and $\mathcal{Y}$. The transformation $\mathbf{R}, \mathbf{t}$ can be estimated by solving

$$\min_{\mathbf{R},\mathbf{t}} = \sum_{\mathbf{x} \in \mathcal{X}_s} \|\mathbf{R}\mathbf{x} + \mathbf{t} - \mathcal{Q}(\mathbf{x})\|_2^2 , \tag{1}$$

where $\mathcal{Q} : \mathcal{X}_s \rightarrow \mathcal{Y}_s$ denotes the function that returns the best matches from set $\mathcal{X}_s$ to set $\mathcal{Y}_s$, with $\mathcal{X}_s$ and $\mathcal{Y}_s$ the selected inlier point sets. Given the matches, Eq. 1 can be solved via SVD [5,18]. The challenging task therefore is to estimate the matching function $\mathcal{Q}$, with only $\mathcal{X}$ and $\mathcal{Y}$ as input.

3.2 Method Overview

Most learning-based 3D registration methods rely on an architecture composed of two modules: the feature extraction module and the point matching module.

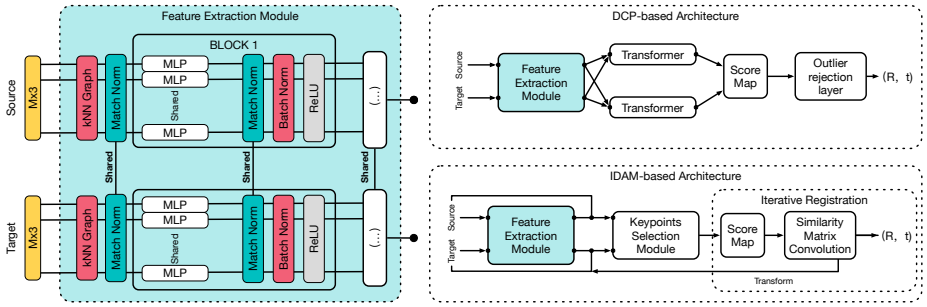

Fig. 2: **Architecture of our DCP-based and IDAM-based frameworks.** We integrate Match Normalization in every block of the feature extraction module, sharing the scale parameter between the corresponding source and target point sets. The feature extraction module is central to the success of 3D registration architectures, such as DCP and IDAM. Its outputs greatly affect the subsequent score maps, and thus the construction of matches and ultimately the pose estimates. Our Match Normalization strategy yields robust features in the presence of real-world data.

The feature extraction module takes the two point sets as input, and outputs a feature vector $\mathbf{f}_x^{(i)}$, resp. $\mathbf{f}_y^{(j)}$, for either each point [61,67] in $\mathcal{X}$, resp. $\mathcal{Y}$, or each keypoint [62,31] extracted from $\mathcal{X}$, resp. $\mathcal{Y}$.

Given these feature vectors, a score map $\mathcal{S} \in \mathbb{R}^{M \times N}$ is formed by computing the similarity between each source-target pair of descriptors. That is, the (i,j)-th element of $\mathcal{S}$ is computed as

$$\mathcal{S}_{i,j} = < \mathbf{f}_x^{(i)}, \mathbf{f}_y^{(j)} >, \quad \forall (i,j) \in [1,M] \times [1,N] \,, \tag{2}$$

where $< \cdot, \cdot >$ is the inner product, and $\mathbf{f}_x^{(i)}, \mathbf{f}_y^{(j)} \in \mathbb{R}^P$. This matrix is then given to the point matching module, whose goal is to find the correct inlier matches between the two point sets while rejecting the outliers.

The parameters of the network are typically trained by minimizing the difference between the ground-truth rotation and translation and those obtained by solving Eq. 1. Because, given predicted matches, the solution to Eq. 1 can be obtained by SVD, training can be achieved by backpropagating the gradient through the SVD operator, following the derivations in [25].

In the remainder of this section, we first introduce our approach to robustifying the feature extraction process, and then discuss the loss function we use to stabilize the training process in the presence of a full rotation range. Finally, we provide the details of the two models, DCPv2 [61] and IDAM [31], in which we implemented our strategies.

3.3 Match Normalization for Robust Feature Extraction

Most learning-based 3D registration methods [61,67,62,68,31] build on the PointNet [42] architecture for the feature extraction module. Specifically, they use

either convolutional layers or MLPs operating on a graph or the raw point set, with the output of each layer being normalized by Batch Normalization followed by ReLU. The normalized features of each layer then go through an additional subnetwork, which aggregates them into global features that are concatenated to the point-wise ones.

In this process, Batch Normalization aims to standardize the feature distributions to speed up training. Batch normalization assumes that every sample follows the same global statistics. In synthetic datasets, where all point clouds have been normalized to a common size, this assumption is typically satisfied. However, real-world data obtained by capturing objects with a depth sensor often includes objects of highly diverse sizes and only depicts unknown portions of these objects. Therefore, the data does not meet the Batch Normalization assumption, and using the same normalization values for all samples within a mini-batch yields a large gap in the distributions of the features extracted by the network. This is illustrated in the top row of Figure 1, where we show the histogram and corresponding probability density function of the output of each layer of a network trained with Batch Normalization. Specifically, for each layer, we show a histogram encompassing all the feature channels for all the points in the source point cloud, and a similar histogram for the corresponding target point cloud. Additional examples are provided in the supplementary material. These plots clearly highlights the differences between the distributions of the corresponding source and target point clouds. In practice, these differences then affect the number of matched points, ultimately leading to low registration accuracy.

To overcome this, we introduce Match Normalization. The central idea between Match Normalization is to force the two point sets to have similar distributions. Specifically, we achieve this by centering the source and target point sets separately but by scaling them with the same parameter. The resulting normalized features then satisfy the Batch Normalization assumption, and, as shown in Figure 2, we then feed them into a Batch Normalization layer to retain the benefit of fast training.

Formally, Match Normalization can be expressed as follows. For a layer with C output channels, let $\mathbf{o}_x \in \mathbb{R}^{C\times M}$ and $\mathbf{o}_y \in \mathbb{R}^{C\times N}$ be the features obtained by processing $\mathcal{X}$ and $\mathcal{Y}$, respectively. We then normalize the features for each point i as

$$\hat{\mathbf{o}}_x^{(i)} = \frac{1}{\beta}(\mathbf{o}_x^{(i)} - \mu_x), \qquad \hat{\mathbf{o}}_y^{(i)} = \frac{1}{\beta}(\mathbf{o}_y^{(i)} - \mu_y), \tag{3}$$

where $\mu_x = \frac{1}{M}\sum_{i=1}^{M} \mathbf{o}_x^{(i)}$, and similarly for μ_y, are calculated separately for $\mathbf{o}_x$ and $\mathbf{o}_y$, but the scale

$$\beta = \max_{(i,j)=(1,1)}^{(M,C)} |\mathbf{o}_x^{(i,j)}| \,, \tag{4}$$

where $\mathbf{o}_x^{(i,j)}$ denotes the j-th feature of the i-th point, is shared by the corresponding source and target point sets. This scale parameter is computed from the source features, which are not subject to partial observations as the source points are sampled from the object model.

Method	Rotation mAP						Translation mAP					
	5°	10°	15°	20°	25°	30°	0.001	0.005	0.01	0.05	0.1	0.5
DCP(v2)+SVD (45°)	0.36	0.75	0.90	0.96	0.98	0.99	0.51	0.92	0.98	1.00	1.00	1.00
DCP(v2)+NLL (45°)	0.72	0.97	1.00	1.00	1.00	1.00	0.54	0.99	1.00	1.00	1.00	1.00
DCP(v2)+SVD (Full)	0.00	0.01	0.02	0.03	0.04	0.05	0.04	0.23	0.42	0.96	1.00	1.00
DCP(v2)+NLL (Full)	0.35	0.67	0.86	0.94	0.97	0.98	0.19	0.57	0.85	1.00	1.00	1.00

Table 1: **Influence of the loss function.** The models are evaluated on the partial-to-partial registration task on ModelNet40 (clean) as in [62,67,31], with either a 45° rotation range, or a full one. For a given rotation range, the NLL loss yields better results than the SVD-based one. In the full rotation range scenario, the SVD-based loss fails completely, while the NLL loss still yields reasonably accurate pose estimates.

One advantage of using the same scale parameter for both point sets is robustness to partial observations and to outliers. Indeed, if the target point cloud had its own scaling parameter, the presence of partial observations, respectively outlier measurements, in the target point cloud might lead to stretching, respectively squeezing, it. By contrast, the source point cloud is complete and does not contain outliers. We thus leverage the intuition that the source and target point sets should be geometrically similar, and use the same scaling parameters in the Match Normalization process.

3.4 NLL Loss Function for Stable Training

In the commonly-used synthetic setting, the relative rotation between the two point clouds is limited to the $[0°, 45°]$ range. By contrast, in real data, the objects' pose may cover the full rotation range. To mimic this, we use the synthetic ModelNet40 dataset and modify the augmentation so as to generate samples in the full rotation range. As shown in Table 1, our DCPv2 baseline, although effective for a limited rotation range, fails in the full range setting. Via a detailed analysis of the training behavior, we traced the reason for this failure back to the choice of loss function. Specifically, the use of an SVD-based loss function yields instabilities in the gradient computation. This is due to the fact that, as can be seen from the mathematical expression of the SVD derivatives in [25], when two singular values are close to each other in magnitude, the derivatives explode.

To cope with this problem, inspired by [50], we propose to use the negative log likelihood loss to impose a direct supervision on the score map. To this end, let $\mathcal{M} \in \{0,1\}^{M \times N}$ be the matrix of ground-truth correspondences, with a 1 indicating a correspondence between a pair of points. To build the ground-truth assignment matrix $\mathcal{M}$, we transform $\mathcal{X}$ using the ground-truth transformation $\mathcal{T}$, giving us $\tilde{\mathcal{X}}$. We then compute the pairwise Euclidean distance matrix between $\tilde{\mathcal{X}}$ and $\mathcal{Y}$, which we threshold to obtain a correspondence matrix $\mathcal{M} \in \{0,1\}$. We augment $\mathcal{M}$ with an extra row and column acting as outlier bins to obtain $\bar{\mathcal{M}}$. The points without any correspondence are treated as outliers, and the corresponding positions in $\bar{\mathcal{M}}$ are set to one. This strategy does not guarantee a bipartite matching, which we address using a forward-backward check.

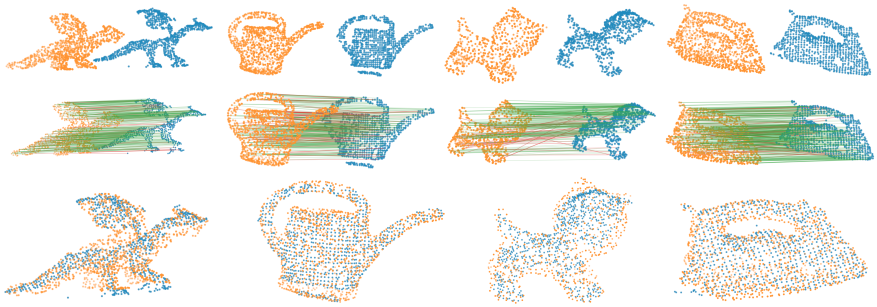

Fig. 3: **Qualitative results of Ours-DCP+ICP. Top:** Input source (orange) and target (blue) point clouds. We show objects from the TUD-L (dragon,watering can), LINEMOD (kitten) and Occluded-LINEMOD (iron) datasets. **Middle:** Matches found by Ours-DCP+ICP, with the true inlier matches in green, and the outlier matches in red. **Bottom:** Registration results, showing that the source and target sets are correctly aligned.

We then express our loss function as the negative log-likelihood

$$\mathcal{L}(\bar{\mathcal{P}}, \bar{\mathcal{M}}) = \frac{-\sum_{i=1}^{M}\sum_{j=1}^{N}(\log \bar{\mathcal{P}}_{i,j})\bar{\mathcal{M}}_{i,j}}{\sum_{i=1}^{M}\sum_{j=1}^{N}\bar{\mathcal{M}}_{i,j}} , \tag{5}$$

where $\bar{\mathcal{P}}$ is the estimated score map, and where the denominator normalizes the loss value so that different training samples containing different number of correspondences have the same influence in the overall empirical risk.

3.5 Network Architectures

In this section, we present the two architectures in which we implemented our strategies. One of them relies on point-wise features whereas the other first extract keypoints, thus illustrating the generality of our contributions.

DCP-based Architecture. In DCPv2, the feature extraction module, denoted by $\Theta(\cdot,\cdot)$, takes two point sets as input, and outputs the feature matrix θ_x, resp. θ_y, i.e., one P-dimensional feature vector per 3D point for $\mathcal{X}$, resp. $\mathcal{Y}$. Then then two feature matrices be passed to a transformer, which learns a function $\phi : \mathbb{R}^{M\times P} \times \mathbb{R}^{N\times P} \rightarrow \mathbb{R}^{M\times P}$, that combines the information of the two point sets. Ultimately, this produces the descriptor matrix $\mathbf{f}_x$, resp. $\mathbf{f}_y$, for $\mathcal{X}$, resp. $\mathcal{Y}$, written as

$$\mathbf{f}_x = \theta_x + \phi(\theta_x, \theta_y), \quad \mathbf{f}_y = \theta_y + \phi(\theta_y, \theta_x) \, . \tag{6}$$

For our architecture, we integrate our Match Normalization strategy to the layers of the feature extractor $\Theta(\cdot,\cdot)$, while keeping the transformer architecture unchanged.

Inspired by previous work [50,67], we choose to use a Sinkhorn layer to handle the partial-to-partial matching case. Specifically, we extend the score matrix $\mathcal{S}$ of Eq. 2 by one row and one column to form an augmented score matrix $\bar{\mathcal{S}}$. The values at the newly-created positions in $\bar{\mathcal{S}}$ are set to

$$\bar{\mathcal{S}}_{i,N+1} = \bar{\mathcal{S}}_{M+1,j} = \bar{\mathcal{S}}_{M+1,N+1} = \alpha, \tag{7}$$

$\forall i \in [1, M],\ \forall j \in [1, N]$, where $\alpha \in \mathbb{R}$ is a fixed parameter, which we set to 1 in practice. The values at the other indices directly come from $\mathcal{S}$. Given the augmented score map $\bar{\mathcal{S}}$, we aim to find a partial assignment $\bar{\mathcal{P}} \in \mathbb{R}^{(M+1)\times(N+1)}$, defining correspondences between the two point sets, extended with the outlier bins. This assignment is obtained by a differentiable version of the Sinkhorn algorithm [67,50], and is used in the calculation of the loss function of Eq. 2. At test time, we use the output of the Sinkhorn layer to find the best set of corresponding points between the two point clouds. In addition to the points found as outliers, we also discard those whose value in the score map are below a threshold.

IDAM-based Architecture. The main difference compared to the DCP-based architecture is that this architecture builds the score map upon selected keypoints instead of all the input points. Similarly to the DCP-based architecture, the IDAM-based one uses a feature extraction module $\Theta(\cdot,\cdot)$. This module can be a traditional local descriptor, FPFH [48], or a learning-based method. We therefore integrate our Match Normalization to the learning-based feature extraction network. The extracted features θ_x and θ_y are then passed to a keypoint selection module, corresponding to a second network that outputs a significance score. The significance score is used to obtain a fixed number of keypoints. We denote the features of the reduced keypoint sets as $\tilde{\theta}_x \in \mathbb{R}^{M'\times P}$ and $\tilde{\theta}_y \in \mathbb{R}^{N'\times P}$. These features, combined with their corresponding original coordinates, are used to calculate a reduced score map $\mathcal{S}' \in \mathbb{R}^{M'\times N'\times(2P+4)}$, which is processed by a similarity matrix convolutional neural network to obtain the final score map used to find the matches. IDAM further incorporates an iterative registration loop to this process to refine the results, as illustrated in Figure 2. Please refer to [31] for more detail.

For IDAM, the loss function is composed of three terms: One to supervise the score matrix as in the DCP-based architecture, and two to supervise the keypoint selection network. The framework assumes that the outliers have been eliminated in the keypoint selection process. Therefore, at test time, we simply compute the argmax of each row to find the best matches. More details can be found in the paper [31].

4 Experiments

4.1 Datasets and Training Parameters

We evaluate our method on three object-level pose estimation real scene datasets: TUD-L [23], LINEMOD [20], Occluded-LINEMOD [7]. The TUD-L dataset contains training and testing image sequences depicting three moving household

objects. LINEMOD, which is the most commonly-used benchmark for object pose estimation, consists of 15 household objects in cluttered scenes. Occluded-LINEMOD is a subset of the LINEMOD dataset, which only contains 8 objects. In contrast with LINEMOD where a single object per image is annotated, Occluded-LINEMOD contains annotations for multiple objects in each image, with severe occlusions between the objects.

For TUD-L, we use the provided real scene training data for training. As there are only 1214 testing images and no explicit training data in Occluded-LINEMOD, we train our network based on the LINEMOD training data. To be specific, we use the PBR dataset provided by the BOP Benchmark [23]. For testing, we follow the BOP 2019 challenge instructions and use the provided testing split for testing. In summary, there are 600 images for TUD-L, 3000 images for LINEMOD and 1445 images for Occluded-LINEMOD.

To obtain the target point clouds from the depth maps, we use the mask provided by the datasets. Note that the resulting point cloud is still partial and may still contain outliers, particularly at the boundary of the object. This step could in principle be achieved by a point cloud or depth map segmentation method, but our results already show that existing frameworks struggle in this scenario, and we therefore leave point cloud segmentation for future work.

We implement our DCP-based pose estimation network in Pytorch [39] and train it from scratch. We use the Adam optimizer [28] with a learning rate of 10^{-3} and mini-batches of size 32, and train the network for $30,000$ iterations. For the OT layer, we use $k = 50$ iterations and set $\lambda = 0.5$. For our IDAM-based architecture, we train the model with the Adam optimizer [28] until convergence, using a learning rate of 10^{-4} and mini-batches of size 32. We use the FPFH implementation from the Open3D [72] library and our custom DGCNN for feature extraction. We set the number of refinement iterations for both the FPFH-based and DGCNN-based versions to 3. For both frameworks, we set the number of points for $\mathcal{X}$ and $\mathcal{Y}$ to be 1024 and 768, respectively, encoding the fact that $\mathcal{Y}$ only contains a visible portion of $\mathcal{X}$. Training was performed on one NVIDIA RTX8000 GPU.

4.2 Evaluation Metrics

For evaluation, in addition to the three metrics used by the BOP benchmark, Visible Surface Discrepancy (VSD) [22,23], Maximum Symmetry-Aware Surface Distance (MSSD) [15] and Maximum Symmetry-Aware Projection Distance (MSPD) [8], we also report the rotation and translation error between the predictions $\hat{R}, \hat{\mathbf{t}}$ and the ground truth $R_{gt}, \mathbf{t}_{gt}$. These errors are computed as

$$E_{rot}(\hat{R}, R_{gt}) = arccos \frac{trace(\hat{R}^\top R_{gt}) - 1}{2} \, , E_{trans}(\hat{\mathbf{t}}, \mathbf{t}_{gt}) = \left\| \hat{\mathbf{t}} - \mathbf{t}_{gt} \right\|_2^2 \, . \quad (8)$$

We summarize the results in terms of mean average precision (mAP) of the estimated relative pose under varying accuracy thresholds, as in [12]. We keep the rotation error unchanged. Furthermore, we also report the ADD metric [63],

(a) Ours-DCP+ICP (b) Ous-IDAM+ICP (c) Super4PCS (d) Teaser++

Fig. 4: Qualitative results on the TUD-L (top), LINEMOD (middle) and Occluded-LINEMOD (bottom) datasets.

which measures the average distance between the 3D model points transformed using the predicted pose and those obtained with the ground-truth one. We set the threshold to be 10% of the model diameter, as commonly done in 6D pose estimation.

4.3 Comparison with Existing Methods

We compare our method to both traditional techniques and learning-based ones. Specifically, for the traditional methods, we used the Open3D [72] implementations of ICP [5] and FGR [71], the official implementation[1] of TEASER++ [64], and the author's binary file[2] for Super4PCS [36]. For DCP, we used the default DCPv2 training settings. However, because of instabilities caused by the SVD computation, we had to train the model several times to eventually find a point before a crash achieving reasonable accuracy. Note that this problem was also reported in [59]. We also tried to train PRNet and RPMNet, but failed to get reasonable results because of similar SVD-related crashes, as also observed in [10]. For IDAM, we report results using both traditional FPFH features and features extracted with a DGCNN. Ours-DCP denotes our approach implemented in the DCPv2 architecture, where we replace the SVD-based loss with the NLL one, replace the softmax layer with a Sinkhorn layer, and integrate Match Normalization with the DGCNN. Ours-IDAM denotes our approach within the IDAM architecture, incorporating Match Normalization in the DGCNN. Since the IDAM does not use an SVD-based loss function, we retained its original loss.

TUD-L dataset. The results of all methods for the TUD-L dataset are summarized in Table 2. Note that the traditional methods based on FPFH features

[1] https://github.com/MIT-SPARK/TEASER-plusplus
[2] https://github.com/nmellado/Super4PCS

Method	Rotation mAP			Translation mAP			ADD	BOP Benchmark			
	5°	10°	20°	1cm	2cm	5cm	0.1d	VSD	MSSD	MSPD	AR
ICP	0.02	0.02	0.02	0.01	0.14	0.57	0.02	0.117	0.023	0.027	0.056
FGR(FPFH)	0.00	0.01	0.01	0.04	0.25	0.63	0.01	0.071	0.007	0.008	0.029
TEASER++(FPFH)	0.13	0.17	0.19	0.03	0.22	0.56	0.17	0.175	0.196	0.193	0.188
Super4PCS	0.30	0.50	0.56	0.05	0.40	0.92	0.54	0.265	0.500	0.488	0.418
DCP(v2)	0.00	0.01	0.02	0.02	0.07	0.55	0.01	0.0253	0.051	0.039	0.038
IDAM(FPFH)	0.05	0.12	0.20	0.03	0.17	0.63	0.13	0.100	0.194	0.166	0.153
IDAM	0.03	0.05	0.10	0.02	0.08	0.49	0.05	0.067	0.108	0.099	0.091
⋆Vidal-Sensors18	-	-	-	-	-	-	-	0.811	0.910	0.907	0.876
⋆Drost	-	-	-	-	-	-	-	0.809	0.875	0.872	0.852
Ours-IDAM	0.36	0.46	0.53	0.23	0.47	0.75	0.46	0.339	0.502	0.492	0.444
Ours-IDAM+ICP	0.56	0.58	0.61	0.55	0.66	0.81	0.58	0.580	0.604	0.618	0.601
Ours-DCP	0.70	0.81	0.87	0.71	0.86	0.97	0.85	0.700	0.853	0.852	0.801
Ours-DCP+ICP	**0.91**	**0.92**	**0.93**	**0.86**	**0.95**	**0.99**	**0.93**	**0.859**	**0.914**	**0.935**	**0.903**

Table 2: Quantitative comparison of our method with previous work on the **TUD-L** real scene dataset. The results for Vidal-Sensor18 [56] and Drost (Drost-CVPR10-3D-Edges) [16] were directly taken from the BOP leaderboard, and, in contrast with all the other results, were obtained without using a mask for the target point cloud. Our contributions allow existing learning-based methods, such as IDAM and DCP, to successfully register real-world data.

yield poor results, because FPFH yields unreliable features in the presence of many smooth areas on the objects. Vanilla DCPv2 and IDAM also struggle with such real-world data. However, these baselines are significantly improved by our Match Normalization strategy, Ours-DCP and Ours-IDAM, both of which outperform Super4PCS. Our results can further be boosted by the use of ICP as a post-processing step.

In the table, we also provide the results of 'Vidal-Sensors18' and 'Drost-CVPR10-3D-Edges', the two best depth-only performers in the BOP leaderboard. Note that these are traditional methods whose results were obtained without using a mask to segment the target point cloud, which makes the comparison favorable to our approach. Nevertheless, our results evidence that our contributions can make learning-based 3D object registration applicable to real-world data, which we believe to be a significant progress in the field.

LINEMOD dataset. In contrast with TUD-L, the LINEMOD dataset contains symmetrical objects and small occlusions at the object boundaries, which increase the difficulty of this dataset. As shown in Table 3, even Super4PCS is therefore unable to yield meaningful results on this dataset. Furthermore, as the LINEMOD training data does not contain any real-world measurements, the training-testing domain gap further complicates the task for learning-based methods. Nevertheless, our approach improves the results of both DCP and IDAM, allowing them to produce reasonably accurate pose estimates.

Occluded-LINEMOD dataset. The Occluded-LINEMOD dataset further increases the challenge compared to LINEMOD by adding severe occlusions, in addition to the still-existing domain gap. As such, as shown in Table 4, the re-

Method	Rotation mAP			Translation mAP			ADD	BOP Benchmark			
	5°	10°	20°	1cm	2cm	5cm	0.1d	VSD	MSSD	MSPD	AR
ICP	0.00	0.01	0.01	0.04	0.27	0.82	0.01	0.092	0.014	0.027	0.044
FGR(FPFH)	0.00	0.00	0.00	0.05	0.31	0.89	0.00	0.068	0.000	0.010	0.026
TEASER++(FPFH)	0.01	0.03	0.05	0.03	0.21	0.73	0.03	0.108	0.076	0.098	0.094
Super4PCS	0.02	0.09	0.15	0.04	0.31	0.89	0.10	0.117	0.178	0.201	0.165
DCP(v2)	0.00	0.00	0.01	0.05	0.24	0.83	0.00	0.057	0.025	0.049	0.044
IDAM(FPFH)	0.00	0.01	0.03	0.03	0.16	0.67	0.01	0.053	0.055	0.069	0.059
IDAM	0.00	0.01	0.05	0.03	0.16	0.71	0.02	0.050	0.070	0.081	0.067
⋆PPF_3D_ICP	-	-	-	-	-	-	-	**0.719**	**0.856**	**0.866**	**0.814**
⋆Drost	-	-	-	-	-	-	-	0.678	0.786	0.789	0.751
Ours-IDAM	0.01	0.07	0.15	0.13	0.38	0.87	0.11	0.148	0.194	0.209	0.184
Ours-IDAM+ICP	0.15	0.23	0.27	0.25	0.54	0.91	0.23	0.352	0.311	0.345	0.336
Ours-DCP	0.10	0.27	0.49	0.26	0.60	0.95	0.37	0.319	0.490	0.529	0.446
Ours-DCP+ICP	**0.43**	**0.59**	**0.67**	**0.49**	**0.83**	**0.97**	**0.60**	0.616	0.680	0.737	0.678

Table 3: Quantitative comparison of our method with previous work on the **LINEMOD** real scene dataset. PPF_3D_ICP [16] and Drost (Drost-CVPR10-3D-Only) [16] are traditional methods and represent the best depth-only performers from the BOP leaderboard.

Method	Rotation mAP			Translation mAP			ADD	BOP Benchmark			
	5°	10°	20°	1cm	2cm	5cm	0.1d	VSD	MSSD	MSPD	AR
ICP	0.01	0.01	0.01	0.07	0.36	0.85	0.01	0.085	0.014	0.032	0.044
FGR(FPFH)	0.00	0.00	0.00	0.08	0.43	0.85	0.00	0.055	0.000	0.009	0.021
TEASER++(FPFH)	0.01	0.02	0.05	0.04	0.26	0.77	0.02	0.096	0.060	0.093	0.083
Super4PCS	0.01	0.03	0.06	0.06	0.31	0.83	0.03	0.054	0.072	0.113	0.080
DCP(v2)	0.00	0.00	0.01	0.03	0.30	0.83	0.00	0.055	0.018	0.059	0.044
IDAM(FPFH)	0.00	0.00	0.02	0.04	0.18	0.73	0.00	0.044	0.033	0.066	0.048
IDAM	0.00	0.02	0.06	0.07	0.26	0.76	0.02	0.063	0.088	0.119	0.090
⋆Vidal-Sensors18	-	-	-	-	-	-	-	0.473	0.625	0.647	0.582
⋆PPF_3D_ICP	-	-	-	-	-	-	-	**0.523**	**0.669**	**0.716**	**0.636**
Ours-IDAM	0.02	0.08	0.18	0.15	0.44	0.84	0.12	0.155	0.204	0.248	0.202
Ours-IDAM+ICP	0.15	0.22	0.32	0.23	0.58	0.88	0.25	0.349	0.320	0.374	0.348
Ours-DCP	0.07	0.19	0.36	0.24	0.57	0.88	0.28	0.263	0.384	0.450	0.365
Ours-DCP+ICP	**0.31**	**0.46**	**0.56**	**0.37**	**0.70**	**0.91**	**0.47**	0.478	0.542	0.612	0.544

Table 4: Quantitative comparison of our method with previous work on the **Occluded-LINEMOD** real scene dataset. Vidal-Sensors18 [56] and PPF_3D_ICP [16] are traditional methods and represent the best depth-only performers from the BOP leaderboard.

sults of all the methods deteriorate. Nevertheless, our approach still allows DCP and IDAM to yield meaningful pose estimates.

4.4 Ablation Study

In this section, we conduct an ablation study to justify the effectiveness of the proposed Match Normalization. Specifically, we evaluate our proposed DCP-based architecture with and without Match Normalization on the TUD-L dataset. In addition to the metrics used in the previous section, we report the number of matches predicted by the network and the number of real inliers within these predicted matches. The predicted matches are those extracted directly from the predicted score map $\bar{\mathcal{P}}$. We set the threshold to identify the true inliers to be 0.02.

Method	Rotation mAP			Translation mAP			ADD	BOP Benchmark				Matches	
	5°	10°	20°	1cm	2cm	5cm	0.1d	VSD	MSSD	MSPD	AR	pred	true
Ours w/o MN	0.21	0.22	0.24	0.22	0.32	0.61	0.23	0.27	0.27	0.27	0.27	24.84	10.44
Ours	0.91	0.92	0.93	0.86	0.95	0.99	0.93	0.86	0.91	0.94	0.90	276.10	262.96

Table 5: **Ablation Study.** We evaluate the influence of our Match Normalization strategy in our DCP-based baseline not only on the same metrics as before but also on the number of matches found by the network.

As shown in Table 5, the number of matches significantly increases with our Match Normalization, and a vast majority of them are true inliers. This evidences that normalizing the source and target point sets with Match Normalization indeed helps the network to find correct matches, and eventually improves the pose estimation performance. Qualitative results obtained with our method are shown in Figure 3.

5 Conclusion

We have identified two factors that prevent the existing learning-based 3D object registration methods from working on real-world data. The first is the gap between the feature distributions of the source and target point sets. The larger the difference between the two distributions, the fewer correct inlier matches are found by the method, which leads to a drop in performance. The second is the use of an SVD-based loss function in the presence of a full rotation range of the target point cloud, which causes instabilities in the gradient computation, and eventually complicates the network's convergence. To cope with the first problem, we have proposed a new normalization method, Match Normalization, which encourages the two point sets to have similar feature distributions by scaling them with a parameter calculated from the source point set. We have addressed the second problem by replacing the SVD-based loss function with a simple yet robust NLL loss function that imposes direct supervision on the score map. Our two contributions are simple, yet effective and general. As such they can be integrated into many learning-based 3D registration frameworks. We have evidenced this by applying them to a DCP-based and an IDAM-based architecture. We have demonstrated the effectiveness of our method on three real-world 6D object pose estimation datasets, TUD-L, LINEMOD and Occluded-LINEMOD. To the best of our knowledge, this is the first time that a learning-based 3D object registration method achieves meaningful results such real-world data. We therefore believe that our strategies will constitute a key component of future point cloud registration frameworks. In the future, we will seek to incorporate a 3D detection module in our framework to jointly segment and estimate the 6D pose of the object of interest.

6 Additional Distribution Visualizations

As mentioned in Section 3.3 of the main paper, we compare the source and target distributions of all the samples in a batch in Figs. 5 to 9. When Match Normal-

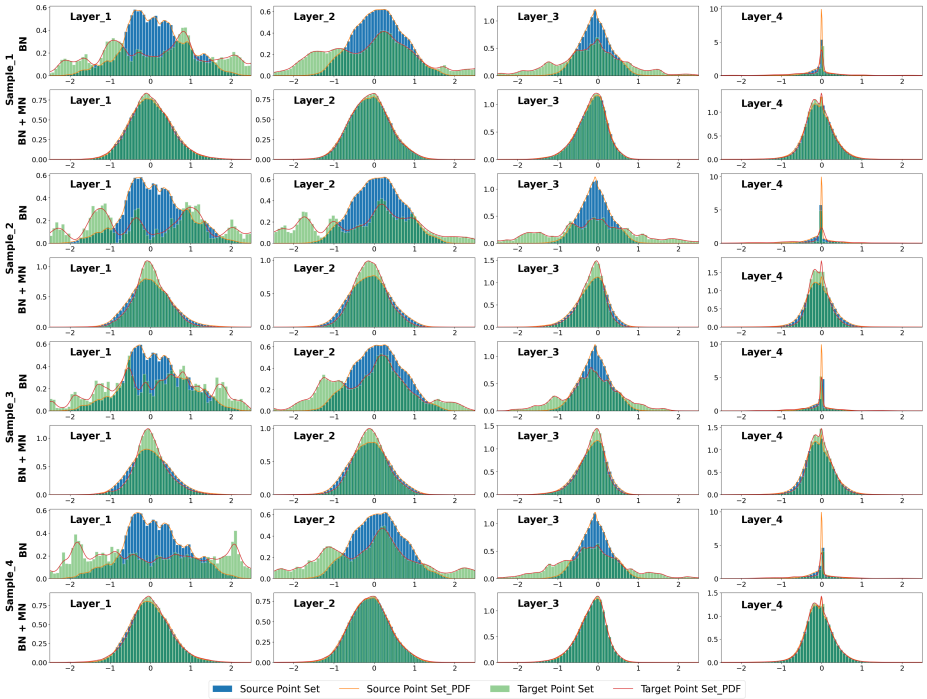

Fig. 5: **Feature distributions at different network layers with real-world data.** We show the distribution of the output of the different layer for different samples within one batch, with and without Match Normalization. Match Normalization consistently makes the source and target distributions much more similar.

ization is not used, there is a clear distribution mismatch in every data pair. These differences then affect the number of matched points, ultimately leading to low registration accuracy. Match Normalization makes the distributions of the two point sets much more similar.

7 Generalization to New Objects and Across Datasets

In this experiment, we demonstrate the model's generalization performance obtained with Match Normalization. To this end, we use a model trained on synthetic data and test it on real datasets. Specifically, for training, we use object mesh models from the ModelNet40 dataset training split. To generate training target point clouds, we exploit Pytorch3D to render the depth map with a randomly given camera pose and intrinsic camera parameters. We use the same parameters as in Section 4.1 to train the model. In the testing phase, we evaluate the resulting model on three real world datasets, TUD-L, LINEMOD and Occluded-LINEMOD, without any fine-tuning. Note that the test objects in these datasets haven't been seen in the training phase. As shown in Tabs. 6

to 8, where we also provide the results obtained by training on real data, our model trained on ModelNet40 only ('Ours-DCP+ICP-M') still outperforms Super4PCS and TEASER++. This demonstrates that Match Normalization makes learning-based 3D object registration applicable to real data while maintaining the ability of point-cloud-based registration to generalize to unseen objects.

Method	Rotation mAP			Translation mAP			ADD	BOP Benchmark			
	5°	10°	20°	1cm	2cm	5cm	0.1d	VSD	MSSD	MSPD	AR
ICP	0.02	0.02	0.02	0.01	0.14	0.57	0.02	0.117	0.023	0.027	0.056
FGR(FPFH)	0.00	0.01	0.01	0.04	0.25	0.63	0.01	0.071	0.007	0.008	0.029
TEASER++(FPFH)	0.13	0.17	0.19	0.03	0.22	0.56	0.17	0.175	0.196	0.193	0.188
Super4PCS	0.30	0.50	0.56	0.05	0.40	0.92	0.54	0.265	0.500	0.488	0.418
DCP(v2)	0.00	0.01	0.02	0.02	0.07	0.55	0.01	0.0253	0.051	0.039	0.038
IDAM(FPFH)	0.05	0.12	0.20	0.03	0.17	0.63	0.13	0.100	0.194	0.166	0.153
IDAM	0.03	0.05	0.10	0.02	0.08	0.49	0.05	0.067	0.108	0.099	0.091
⋆Vidal-Sensors18	-	-	-	-	-	-	-	0.811	0.910	0.907	0.876
⋆Drost	-	-	-	-	-	-	-	0.809	0.875	0.872	0.852
Ours-IDAM	0.36	0.46	0.53	0.23	0.47	0.75	0.46	0.339	0.502	0.492	0.444
Ours-IDAM+ICP	0.56	0.58	0.61	0.55	0.66	0.81	0.58	0.580	0.604	0.618	0.601
Ours-DCP	0.70	0.81	0.87	0.71	0.86	0.97	0.85	0.700	0.853	0.852	0.801
Ours-DCP+ICP	**0.91**	**0.92**	**0.93**	**0.86**	**0.95**	**0.99**	**0.93**	**0.859**	**0.914**	**0.935**	**0.903**
Ours-DCP+ICP-M	0.62	0.66	0.72	0.60	0.73	0.89	0.70	0.666	0.706	0.725	0.699

Table 6: Quantitative comparison of our method with previous work on the **TUD-L** real scene dataset. The results for Vidal-Sensor18 [56] and Drost (Drost-CVPR10-3D-Edges) [16] were directly taken from the BOP leaderboard, and, in contrast with all the other results, were obtained without using a mask for the target point cloud. 'Ours-DCP+ICP-M' is the model which trained on the synthetic dataset ModelNet40. Our contributions not only allow existing learning-based methods to successfully register real-world data, but also keep their generalization ability to unseen objects.

References

1. Agamennoni, G., Fontana, S., Siegwart, R.Y., Sorrenti, D.G.: Point clouds registration with probabilistic data association. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4092–4098. IEEE (2016) 3
2. Aiger, D., Mitra, N.J., Cohen-Or, D.: 4-points congruent sets for robust pairwise surface registration. In: ACM SIGGRAPH 2008 papers. pp. 1–10 (2008) 3
3. Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S.: Pointnetlk: Robust & efficient point cloud registration using pointnet. In: Conference on Computer Vision and Pattern Recognition. pp. 7163–7172. Long Beach, California (2019) 1, 3
4. Atzmon, M., Maron, H., Lipman, Y.: Point convolutional neural networks by extension operators. In: ACM Transactions on Graphics (TOG). TOG (2018) 1
5. Besl, P., Mckay, N.: A method for registration of 3d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(2), 239–256 (February 1992) 3, 4, 11
6. Bouaziz, S., Tagliasacchi, A., Pauly, M.: Sparse iterative closest point. In: Computer graphics forum. vol. 32, pp. 113–123. Wiley Online Library, Hoboken, New Jersey (2013) 3

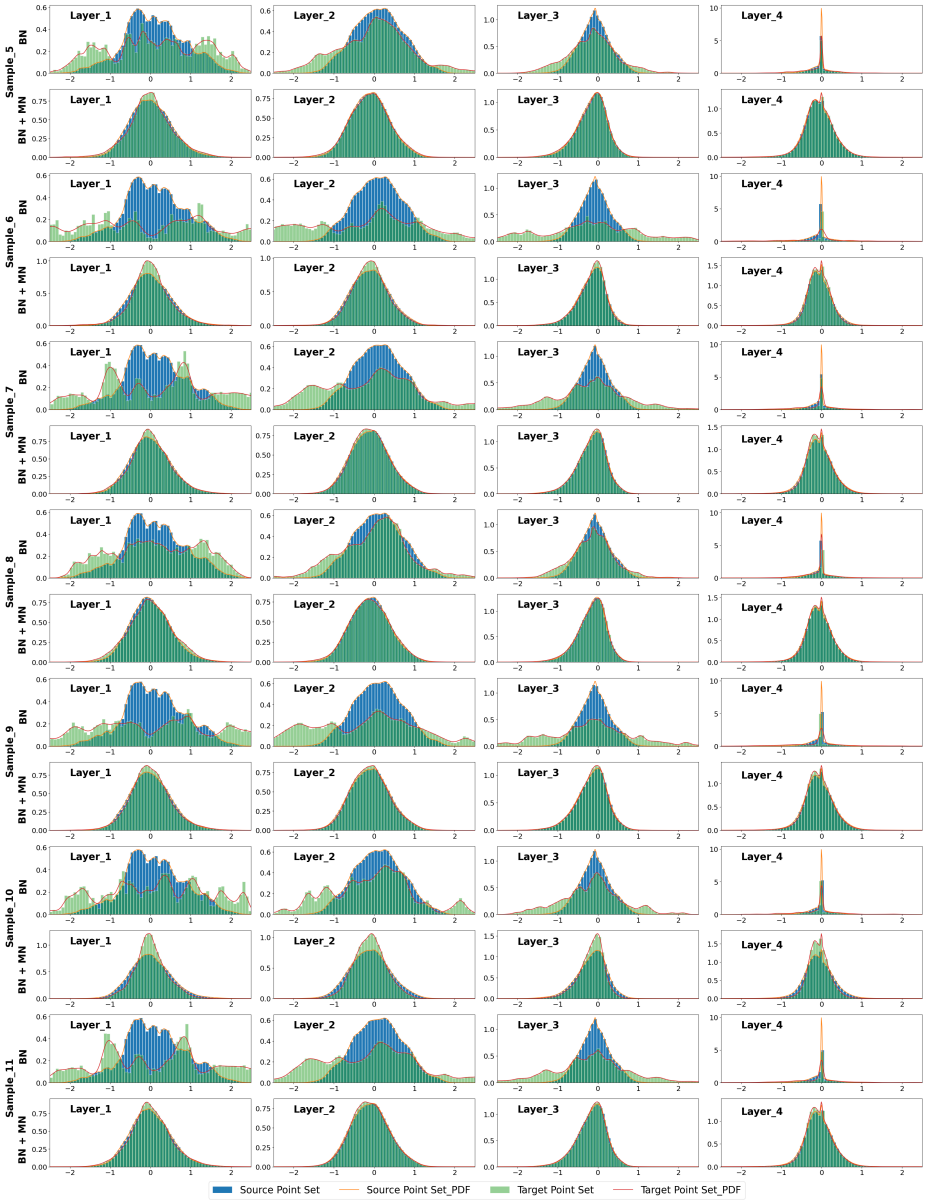

Fig. 6: **Feature distributions at different network layers with real-world data.**

7. Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6d object pose estimation using 3d object coordinates. In: European conference on computer vision. pp. 536–551. Springer, Zürich, Switzerland (2014) 1, 2, 3, 9
8. Brachmann, E., Michel, F., Krull, A., Yang, M.Y., Gumhold, S., et al.: Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In: Pro-

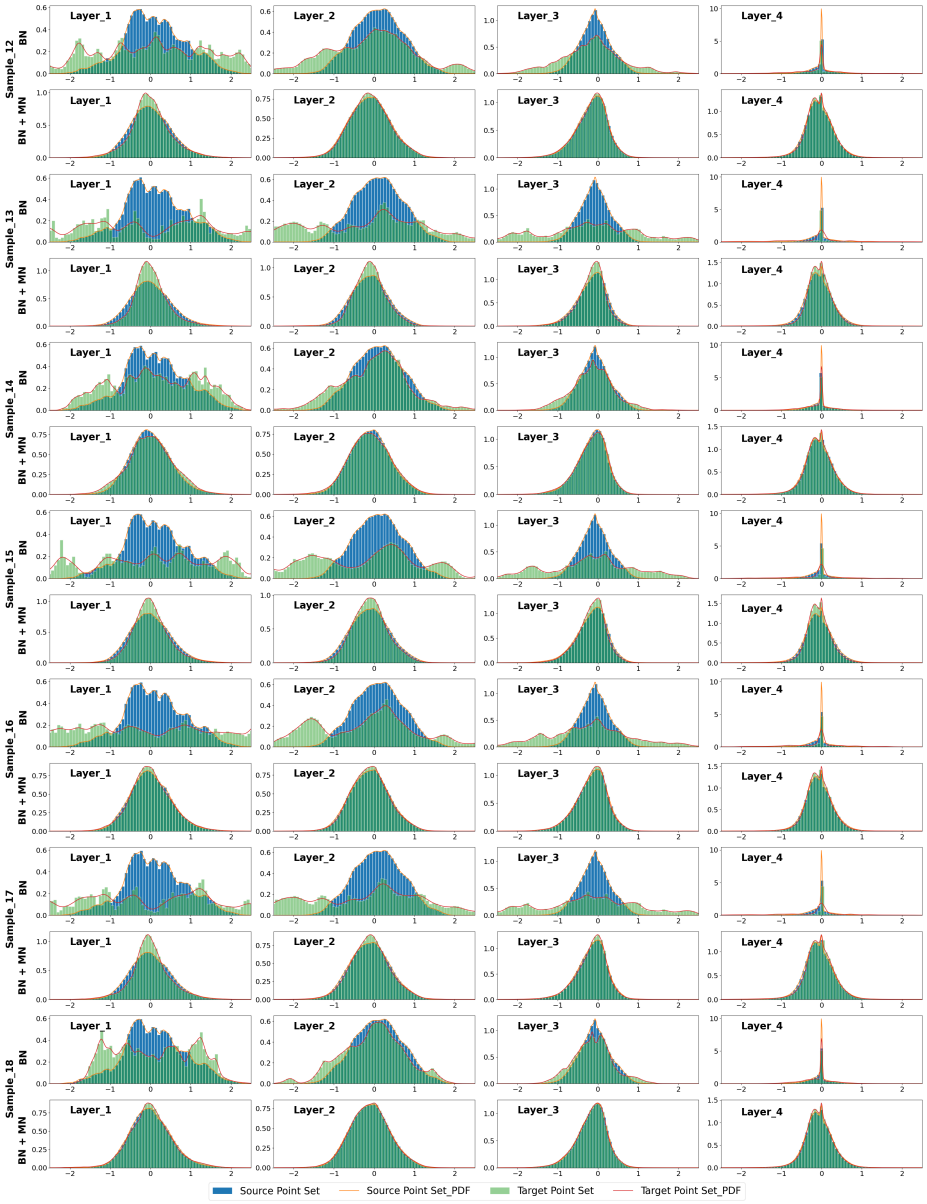

Fig. 7: **Feature distributions at different network layers with real-world data.**

ceedings of the IEEE conference on computer vision and pattern recognition. pp. 3364–3372 (2016) 10

9. Cao, A.Q., Puy, G., Boulch, A., Marlet, R.: Pcam: Product of cross-attention matrices for rigid registration of point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13229–13238 (2021) 2

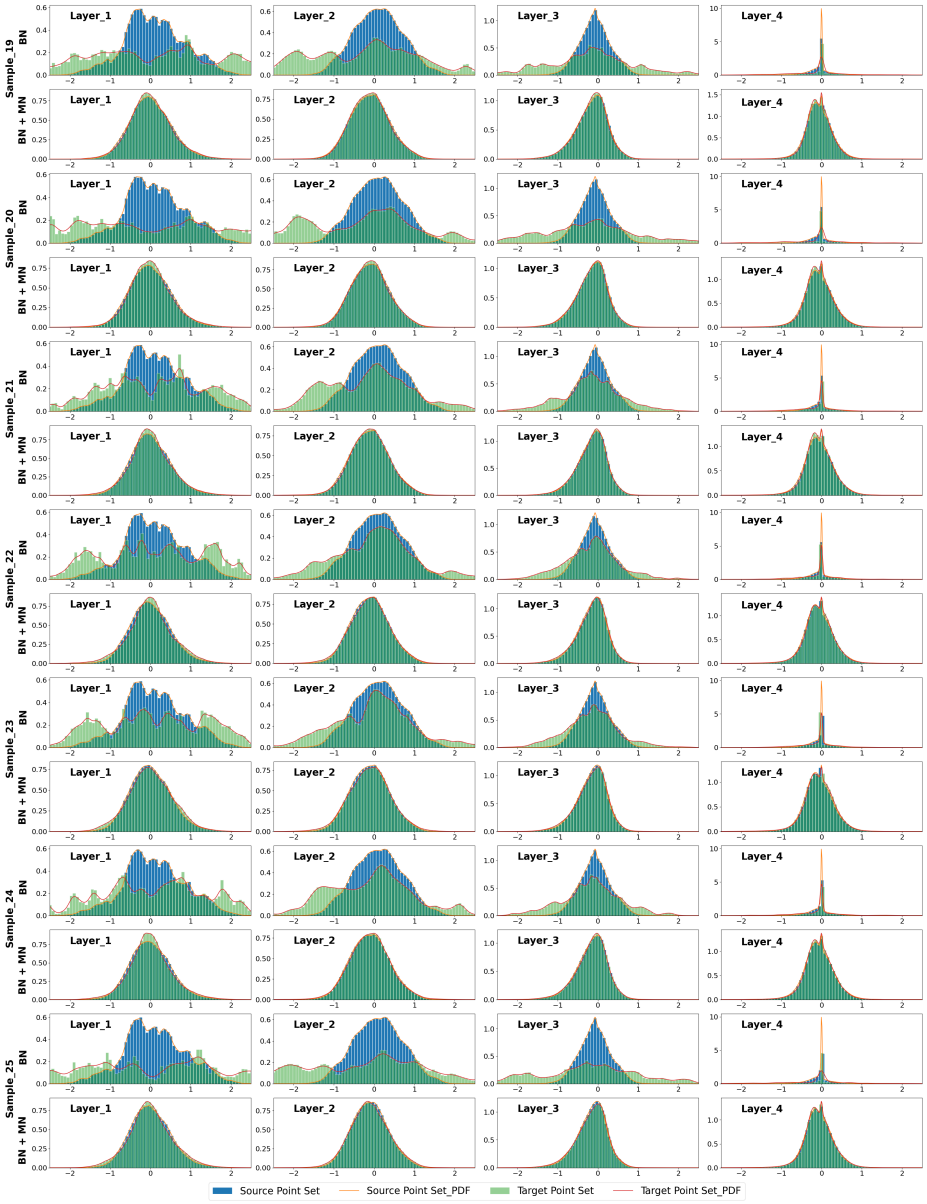

Fig. 8: **Feature distributions at different network layers with real-world data.**

10. Choy, C., Dong, W., Koltun, V.: Deep global registration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Online (2020) 2, 11

11. Choy, C., Park, J., Koltun, V.: Fully convolutional geometric features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8958–

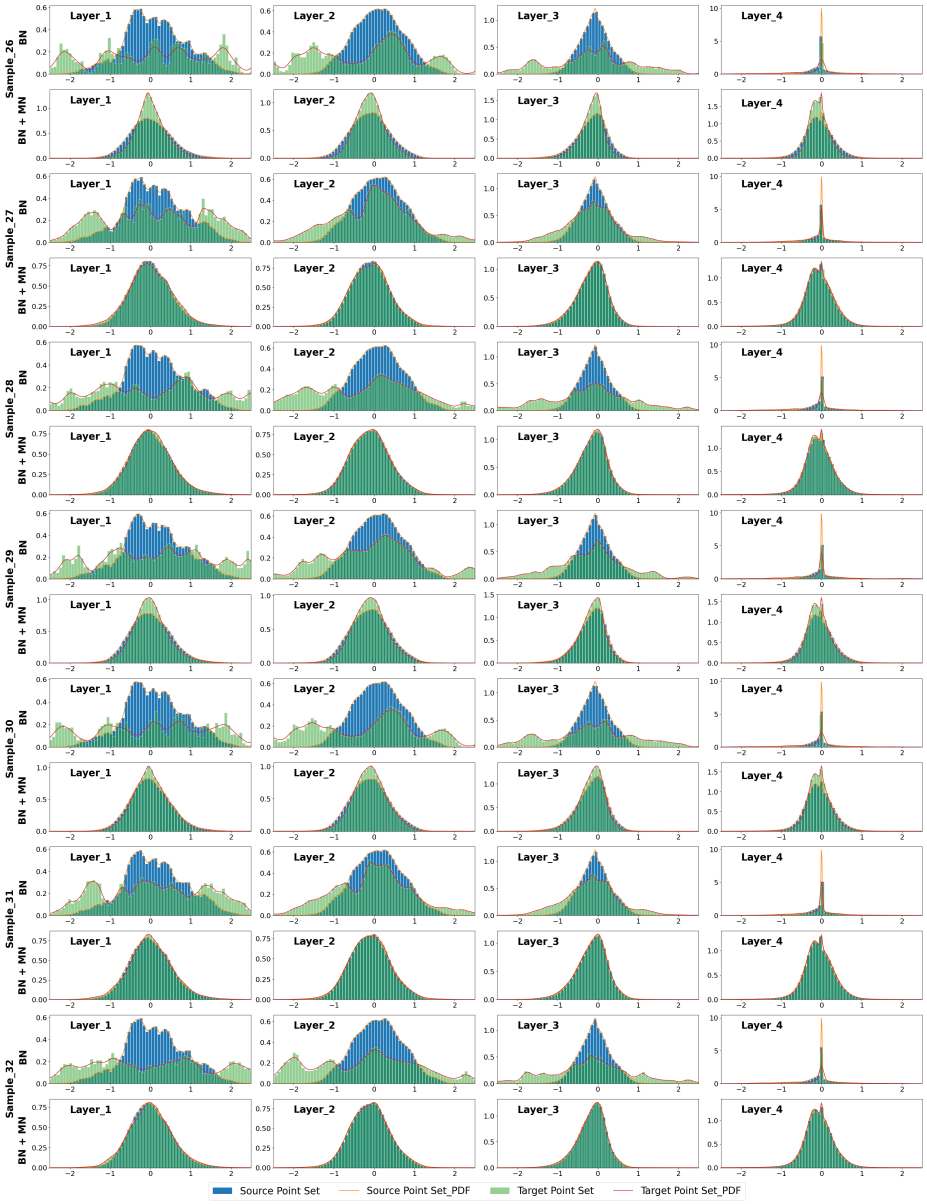

Fig. 9: **Feature distributions at different network layers with real-world data.**

8966 (2019) 2

12. Dang, Z., Moo Yi, K., Hu, Y., Wang, F., Fua, P., Salzmann, M.: Eigendecomposition-free training of deep networks with zero eigenvalue-based losses. In: European Conference on Computer Vision. pp. 768–783. Munich, Germany (2018) 10

Method	Rotation mAP 5°	10°	20°	Translation mAP 1cm	2cm	5cm	ADD 0.1d	BOP Benchmark VSD	MSSD	MSPD	AR
ICP	0.01	0.01	0.01	0.07	0.36	0.85	0.01	0.085	0.014	0.032	0.044
FGR(FPFH)	0.00	0.00	0.00	0.08	0.43	0.85	0.00	0.055	0.000	0.009	0.021
TEASER++(FPFH)	0.01	0.02	0.05	0.04	0.26	0.77	0.02	0.096	0.060	0.093	0.083
Super4PCS	0.01	0.03	0.06	0.06	0.31	0.83	0.03	0.054	0.072	0.113	0.080
DCP(v2)	0.00	0.00	0.01	0.03	0.30	0.83	0.00	0.055	0.018	0.059	0.044
IDAM(FPFH)	0.00	0.00	0.02	0.04	0.18	0.73	0.00	0.044	0.033	0.066	0.048
IDAM	0.00	0.02	0.06	0.07	0.26	0.76	0.02	0.063	0.088	0.119	0.090
⋆Vidal-Sensors18	-	-	-	-	-	-	-	0.473	0.625	0.647	0.582
⋆PPF_3D_ICP	-	-	-	-	-	-	-	**0.523**	**0.669**	**0.716**	**0.636**
Ours-IDAM	0.02	0.08	0.18	0.15	0.44	0.84	0.12	0.155	0.204	0.248	0.202
Ours-IDAM+ICP	0.15	0.22	0.32	0.23	0.58	0.88	0.25	0.349	0.320	0.374	0.348
Ours-DCP	0.07	0.19	0.36	0.24	0.57	0.88	0.28	0.263	0.384	0.450	0.365
Ours-DCP+ICP	**0.31**	**0.46**	**0.56**	**0.37**	**0.70**	**0.91**	**0.47**	0.478	0.542	0.612	0.544
Ours-DCP+ICP-M	0.18	0.25	0.32	0.26	0.57	0.86	0.27	0.343	0.328	0.393	0.354

Table 7: Quantitative comparison of our method with previous work on the **Occluded-LINEMOD** real scene dataset. Vidal-Sensors18 [56] and PPF_3D_ICP [16] are traditional methods and represent the best depth-only performers from the BOP leaderboard.

Method	Rotation mAP 5°	10°	20°	Translation mAP 1cm	2cm	5cm	ADD 0.1d	BOP Benchmark VSD	MSSD	MSPD	AR
ICP	0.00	0.01	0.01	0.04	0.27	0.82	0.01	0.092	0.014	0.027	0.044
FGR(FPFH)	0.00	0.00	0.00	0.05	0.31	0.89	0.00	0.068	0.000	0.010	0.026
TEASER++(FPFH)	0.01	0.03	0.05	0.03	0.21	0.73	0.03	0.108	0.076	0.098	0.094
Super4PCS	0.02	0.09	0.15	0.04	0.31	0.89	0.10	0.117	0.178	0.201	0.165
DCP(v2)	0.00	0.00	0.01	0.05	0.24	0.83	0.00	0.057	0.025	0.049	0.044
IDAM(FPFH)	0.00	0.01	0.03	0.03	0.16	0.67	0.01	0.053	0.055	0.069	0.059
IDAM	0.00	0.01	0.05	0.03	0.16	0.71	0.02	0.050	0.070	0.081	0.067
⋆PPF_3D_ICP	-	-	-	-	-	-	-	**0.719**	**0.856**	**0.866**	**0.814**
⋆Drost	-	-	-	-	-	-	-	0.678	0.786	0.789	0.751
Ours-IDAM	0.01	0.07	0.15	0.13	0.38	0.87	0.11	0.148	0.194	0.209	0.184
Ours-IDAM+ICP	0.15	0.23	0.27	0.25	0.54	0.91	0.23	0.352	0.311	0.345	0.336
Ours-DCP	0.10	0.27	0.49	0.26	0.60	0.95	0.37	0.319	0.490	0.529	0.446
Ours-DCP+ICP	**0.43**	**0.59**	**0.67**	**0.49**	**0.83**	**0.97**	**0.60**	0.616	0.680	0.737	0.678
Ours-DCP+ICP-M	0.25	0.35	0.42	0.34	0.66	0.93	0.36	0.449	0.446	0.499	0.465

Table 8: Quantitative comparison of our method with previous work on the **LINEMOD** real scene dataset. PPF_3D_ICP [16] and Drost (Drost-CVPR10-3D-Only) [16] are traditional methods and represent the best depth-only performers from the BOP leaderboard.

13. Deng, H., Birdal, T., Ilic, S.: Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 602–618 (2018) 2
14. Deng, H., Birdal, T., Ilic, S.: Ppfnet: Global context aware local features for robust 3d point matching. In: Conference on Computer Vision and Pattern Recognition. pp. 195–205. Salt Lake City, Utah (2018) 2
15. Drost, B., Ulrich, M., Bergmann, P., Hartinger, P., Steger, C.: Introducing mvtec itodd-a dataset for 3d object recognition in industry. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 2200–2208. Venice, Italy (2017) 10
16. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3d object recognition. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 998–1005 (2010) 3, 12, 13, 16, 21

17. Fitzgibbon, A.W.: Robust registration of 2d and 3d point sets. Image and vision computing **21**(13-14), 1145–1153 (2003) 3
18. Gower, J.C.: Generalized procrustes analysis. Psychometrika **40**(1), 33–51 (1975) 4
19. Hähnel, D., Burgard, W.: Probabilistic matching for 3d scan registration. In: Proc. of the VDI-Conference Robotik. vol. 2002. Citeseer (2002) 3
20. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: Asian Conference on Computer Vision. pp. 548–562. Daejeon (2012) 1, 2, 3, 9
21. Hinzmann, T., Stastny, T., Conte, G., Doherty, P., Rudol, P., Wzorek, M., Galceran, E., Siegwart, R., Gilitschenski, I.: Collaborative 3d reconstruction using heterogeneous uavs: System and experiments. In: International Symposium on Experimental Robotics. pp. 43–56. Springer (2016) 3
22. Hodaň, T., Matas, J., Obdržálek, Š.: On evaluation of 6d object pose estimation. In: European Conference on Computer Vision. pp. 606–619. Springer (2016) 10
23. Hodan, T., Michel, F., Brachmann, E., Kehl, W., GlentBuch, A., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X., et al.: Bop: Benchmark for 6D Object Pose Estimation. In: European Conference on Computer Vision. pp. 19–34. Munich, Germany (2018) 1, 2, 3, 9, 10
24. Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., Schindler, K.: Predator: Registration of 3d point clouds with low overlap. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4267–4276 (2021) 2
25. Ionescu, C., Vantzos, O., Sminchisescu, C.: Matrix backpropagation for deep networks with structured layers. In: Conference on Computer Vision and Pattern Recognition. Boston, MA, USA (2015) 5, 7
26. Izatt, G., Dai, H., Tedrake, R.: Globally optimal object pose estimation in point clouds with mixed-integer programming. In: Robotics Research. pp. 695–710. Springer, Ventura, CA (2020) 3
27. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. TPAMI **21**(5), 433–449 (1999) 3
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations. San Diego, CA, USA (2015) 10
29. Labbé, Y., Carpentier, J., Aubry, M., Sivic, J.: Cosypose: Consistent multi-view multi-object 6d pose estimation. In: European Conference on Computer Vision. pp. 574–591. Springer, Online (2020) 1
30. Le, H.M., Do, T.T., Hoang, T., Cheung, N.M.: Sdrsac: Semidefinite-based randomized approach for robust point cloud registration without correspondences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 124–133 (2019) 3
31. Li, J., Zhang, C., Xu, Z., Zhou, H., Zhang, C.: Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In: ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16. pp. 378–394. Springer (2020) 4, 5, 7, 9
32. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. In: Advances in Neural Information Processing Systems. pp. 820–830. Montréal, Quebec, Canada (2018) 1
33. Li, Y., Wang, G., Ji, X., Xiang, Y., Fox, D.: DeepIM: Deep Iterative Matching for 6D Pose Estimation. In: European Conference on Computer Vision. pp. 683–698. Munich, Germany (2018) 1

34. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence. Vancouver, British Columb (1981) 3
35. Maron, H., Dym, N., Kezurer, I., Kovalsky, S., Lipman, Y.: Point registration via efficient convex relaxation. ACM Transactions on Graphics (TOG) **35**(4), 1–12 (2016) 3
36. Mellado, N., Aiger, D., Mitra, N.J.: Super 4pcs fast global pointcloud registration via smart indexing. In: Computer graphics forum. vol. 33, pp. 205–215. Wiley Online Library (2014) 3, 11
37. Mohamad, M., Ahmed, M.T., Rappaport, D., Greenspan, M.: Super generalized 4pcs for 3d registration. In: 2015 International Conference on 3D Vision. pp. 598–606. IEEE (2015) 3
38. Park, K., Patten, T., Vincze, M.: Pix2pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation. In: International Conference on Computer Vision. pp. 7668–7677. Seoul, Korea (2019) 1
39. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: International Conference on Learning Representations. Toulon, France (2017) 10
40. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: PVNet: Pixel-wise Voting Network for 6DoF Pose Estimation. In: Conference on Computer Vision and Pattern Recognition. pp. 4561–4570. Long Beach, California (2019) 1
41. Pomerleau, F., Colas, F., Siegwart, R., et al.: A review of point cloud registration algorithms for mobile robotics. Foundations and Trends® in Robotics **4**(1), 1–104 (2015) 3
42. Qi, C., Su, H., Mo, K., Guibas, L.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Conference on Computer Vision and Pattern Recognition. Honolulu, Hawaii (2017) 1, 3, 5
43. Qi, C., Yi, L., Su, H., Guibas, L.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems. Long Beach, California, United States (2017) 1
44. Rad, M., Lepetit, V.: Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In: International Conference on Computer Vision. pp. 3828–3836. Venice, Italy (2017) 1
45. Raposo, C., Barreto, J.P.: Using 2 point+ normal sets for fast registration of point clouds with small overlap. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 5652–5658. IEEE (2017) 3
46. Rosen, D.M., Carlone, L., Bandeira, A.S., Leonard, J.J.: Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group. The International Journal of Robotics Research **38**(2-3), 95–125 (2019) 3
47. Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: Proceedings Third International Conference on 3-D Digital Imaging and Modeling. pp. 145–152. IEEE, Quebec City, Canada (2001) 3
48. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: International Conference on Robotics and Automation. pp. 3212–3217. IEEE, Kobe, Japan (2009) 3, 9
49. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Aligning point cloud views using persistent feature histograms. In: International Conference on Intelligent Robots and Systems. pp. 3384–3391. IEEE, Nice, France (2008) 3
50. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Conference on Computer Vision and Pattern Recognition. IEEE, Long Beach, California (2019) 7, 9

51. Segal, A., Haehnel, D., Thrun, S.: Generalized-icp. In: In Robotics: Science and Systems. Cambridge (2009) 3
52. Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M.H., Kautz, J.: Splatnet: Sparse lattice networks for point cloud processing. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2530–2539 (2018) 1
53. Sundermeyer, M., Marton, Z.C., Durner, M., Brucker, M., Triebel, R.: Implicit 3D Orientation Learning for 6D Object Detection from RGB Images. In: European Conference on Computer Vision. pp. 699–715. Munich, Germany (2018) 1
54. Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., Birchfield, S.: Deep object pose estimation for semantic robotic grasping of household objects. arXiv preprint arXiv:1809.10790 (2018) 1
55. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008. Long Beach, California, United States (2017) 3
56. Vidal, J., Lin, C.Y., Lladó, X., Martí, R.: A method for 6d pose estimation of free-form rigid objects using point pair features on range data. Sensors **18**(8), 2678 (2018) 3, 12, 13, 16, 21
57. Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S.: Densefusion: 6D Object Pose Estimation by Iterative Dense Fusion. In: Conference on Computer Vision and Pattern Recognition. pp. 3343–3352. Long Beach, California (2019) 1
58. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation. In: International Conference on Computer Vision. pp. 2642–2651. Seoul, Korea (2019) 1
59. Wang, W., Dang, Z., Hu, Y., Fua, P., Salzmann, M.: Backpropagation-friendly eigendecomposition. In: Advances in Neural Information Processing Systems. pp. 3156–3164. Vancouver, British Columbia, Canada (2019) 11
60. Wang, Y., Sun, Y., Liu, Z., Sarma, S., Bronstein, M., Solomon, J.: Dynamic graph cnn for learning on point clouds. In: ACM Transactions on Graphics (TOG). TOG (2019) 1, 3
61. Wang, Y., Solomon, J.M.: Deep closest point: Learning representations for point cloud registration. In: International Conference on Computer Vision. pp. 3523–3532. Seoul, Korea (2019) 1, 2, 3, 5
62. Wang, Y., Solomon, J.M.: Prnet: Self-supervised learning for partial-to-partial registration. In: Advances in Neural Information Processing Systems. pp. 8812–8824. Vancouver, British Columbia, Canada (2019) 1, 2, 4, 5, 7
63. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In: Robotics: Science and Systems Conference. Pittsburgh, PA, USA (2018) 10
64. Yang, H., Carlone, L.: A polynomial-time solution for robust registration with extreme outlier rates. In: Robotics: Science and Systems Conference. Freiburg im Breisgau, Germany (2019) 3, 11
65. Yang, H., Shi, J., Carlone, L.: Teaser: Fast and certifiable point cloud registration. In: arXiv Preprint (2020) 3
66. Yang, J., Li, H., Campbell, D., Jia, Y.: Go-icp: A globally optimal solution to 3d icp point-set registration. TPAMI **38**(11), 2241–2254 (2015) 3

67. Yew, Z.J., Lee, G.H.: Rpm-net: Robust point matching using learned features. In: Conference on Computer Vision and Pattern Recognition. Online (2020) 1, 2, 4, 5, 7, 9
68. Yuan, W., Eckart, B., Kim, K., Jampani, V., Fox, D., Kautz, J.: Deepgmr: Learning latent gaussian mixture models for registration. In: European Conference on Computer Vision. pp. 733–750. Springer (2020) 1, 4, 5
69. Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R.R., Smola, A.J.: Deep sets. In: Advances in Neural Information Processing Systems. pp. 3391–3401. Long Beach, California, United States (2017) 1, 3
70. Zakharov, S., Shugurov, I., Ilic, S.: DPOD: 6D Pose Object Detector and Refiner. In: International Conference on Computer Vision. Seoul, Korea (2019) 1
71. Zhou, Q.Y., Park, J., Koltun, V.: Fast global registration. In: European Conference on Computer Vision. pp. 766–782. Springer, Amsterdam, the Netherlands (2016) 3, 11
72. Zhou, Q.Y., Park, J., Koltun, V.: Open3d: A modern library for 3d data processing. In: arXiv Preprint (2018) 10, 11