2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

SD-Net: Symmetric-Aware Keypoint Prediction and Domain Adaptation for 6D Pose Estimation In Bin-picking Scenarios

Ding-Tao Huang[1,2,†], En-Te Lin[1,†], Lipeng Chen[2,†], Li-Fu Liu[1], Long Zeng[1*]

***Abstract*— Despite the success of 6D pose estimation in bin-picking scenarios, existing methods still struggle to produce accurate prediction results for symmetry objects in real-world scenarios. The primary bottlenecks include 1) the ambiguity in keypoints caused by object symmetries; and 2) the domain gap between real and synthetic data. To circumvent these problems, we propose a novel 6D pose estimation network with symmetric-aware keypoint prediction and self-training domain adaptation (SD-Net). SD-Net builds on point-wise keypoint regression and deep hough voting to perform reliable keypoint detection under clutter and occlusion. Specifically, at the keypoint prediction stage, we propose a robust 3D keypoint selection strategy considering the symmetry class of objects and equivalent keypoints, which facilitate locating 3D keypoints even in highly occluded scenes. Additionally, we build an effective filtering algorithm on predicted keypoints to dynamically eliminate multiple ambiguity and outlier keypoint candidates. At the domain adaptation stage, we propose the self-training framework using a student-teacher training scheme. To carefully distinguish reliable predictions, we harness tailored heuristics for 3D geometry pseudo labelling based on semi-chamfer distance. On the public Siléane dataset, SD-Net achieves state-of-the-art results, obtaining an average precision of 96%. Testing learning and generalization abilities on public Parametric datasets, SD-Net is 8% higher than the state-of-the-art method.**

I. Introduction

The estimation of 6D object pose is an essential prerequisite for robotic tasks such as grasping and manipulation, especially in bin-picking scenarios [1]. Recent studies that employ learning-based techniques have shown promising results for this particular task [2]. These methods primarily fall into two categories: holistic methods [1], [3] and keypoint-based methods [4], [5]. Keypoint-based methods employ intermediate variables to predict the 6D object pose, effectively circumventing the nonlinear rotation space and providing a promising direction for the exploration of 6D object pose estimation [6].

Nonetheless, two major challenges persist in hindering the estimation performance of keypoint-based pose estimation approaches, as illustrated in Fig. 1. The first challenge lies in the absence of a robust strategy for selecting keypoints during the keypoint prediction stage. This deficiency is particularly evident when dealing with objects where keypoints on symmetric components are hardly distinguishable. Many previous works [7], [8] have chosen keypoints from a subset of the object's bounding box (BBox) corners which are away from the surface of objects. Some other works [4], [5] employ the farthest point sampling (FPS) algorithm to sample keypoints on the object's surface according to their relative proximity. ParametricNet [9] utilizes a keypoint selection strategy based on object parameters. Despite promising results, these methods fail to consider object geometric symmetry characteristics. The symmetry of an object can result in multiple points on the surface that have similar geometric features to the selected keypoints. These ambiguity points can mislead the network in predicting keypoints.

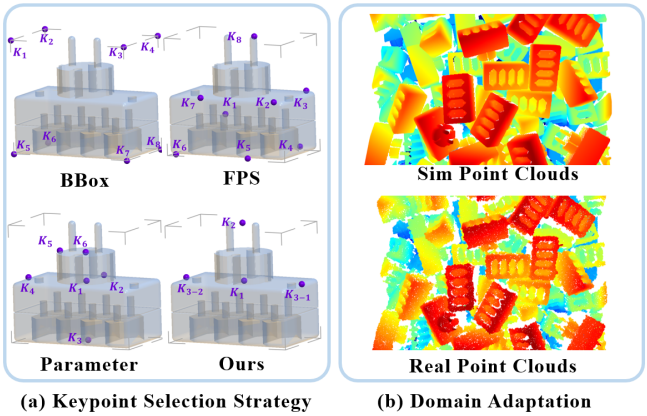


Fig. 1. This work addresses two major problems in 6D pose estimation. (a) The keypoints sampled by BBox, FPS, Parameter and our strategy. BBox, FPS and Parameter keypoints ignore object geometric symmetry characteristics, while our strategy adaptively selects keypoints based on object symmetry class. (b) Existing methods typically exhibit inferior performance when applied to real-world point clouds, due to the persistent domain gap between real and synthetic data point clouds.

Secondly, annotating 6D object poses in the real world is labour-intensive. As a result, the power of simulation is often harnessed to virtually generate 6D pose labels for training deep learning models [10]. However, these methods typically exhibit inferior performance when applied to real-world data, due to the persistent domain gap between real and synthetic data, as demonstrated in Fig. 1. Many approaches [11], [12] rely on a customized simulation strategy to produce high-quality synthetic data, while others [13], [14] necessitate a specialized network architecture to extract domain-invariant signals. Alternatively, some methods employ render-and-compare techniques [10], [15] for self-supervision on unlabeled real data. These approaches utilize 2D appearance information of objects to distinguish reliable predictions.

† Equal contribution.

This research was funded by the National Natural Science Foundation of China (Grant No. 61972220) and Guangdong Natural Science Fund-General Programme Grand No. 2022A1515011234.

1 Tsinghua Shenzhen International Graduate School, SZ, China. Dingtao Huang conducted this work at Tencent Robotics X during his internship. {hdt22, linet22, llf23}@mails.tsinghua.edu.cn
2 Tencent Robotics X, SZ, China. lipengchen@tencent.com
* Corresponding author. zenglong@sz.tsinghua.edu.cn

Nevertheless, texture-less objects provide essentially no 2D effective features, posing a challenge for these methods.

In this study, we introduce a **S**ymmetric-aware keypoint prediction and **D**omain adaptation **Net**work (SD-Net) for 6D object pose estimation in bin-picking scenarios. SD-Net is constructed based on point-wise keypoint regression and deep hough voting. The inclusion of a voting mechanism equips our model with the capability to perform reliable keypoint detection in bin-picking scenarios. To solve the issue of object symmetry, we propose a new keypoint selection and filtering algorithm when performing keypoint prediction. Subsequently, all equivalent keypoints are calculated according to equivalent rotation matrices. This selection method, which takes into account object symmetry class, significantly streamlines the network's task of location and bolsters the pose estimation performance. We also implement a keypoint filtering algorithm to choose the predicted keypoints with the highest confidence before generating pose hypotheses.

In addressing the domain gap, we propose a sim-to-real framework under a student-teacher learning scheme which can be generalized to texture-less objects. We initially train the teacher model in a fully-supervised manner with abundant synthesized data. Subsequently, we use the teacher model to generate pseudo labels on real-world data, and then these pseudo labels are used to update the student network. To facilitate robustness, we propose a tailored heuristic for 3D geometry pseudo labelling that relies on semi-chamfer distance, enabling the careful identification of reliable predictions. Moreover, the integration of mask labels contributes to the stability of the training process.

We benchmark our proposed method using the Siléane [16] and Parametric [9] datasets. Experimental results demonstrate that SD-Net outperforms state-of-the-art methods. On the Siléane dataset, SD-Net achieves a 6% improvement in average precision. Meanwhile, on the Parametric dataset, SD-Net surpasses the state-of-the-art method by 8% in terms of average precision. In summary, the main contributions of this work are:

- We introduce a novel 6D object pose estimation network with symmetric-aware keypoint prediction and domain adaptation, which achieves state-of-the-art estimation performance on the Siléane and Parametric datasets.
- We propose a new keypoint selection method that considers object symmetry class and a robust keypoint filtering algorithm that dynamically eliminates multiple and outlier keypoint candidates.
- We propose an iterative self-training framework for domain adaptation in 6D object pose estimation, which leverages the 3D geometry information of objects to carefully distinguish pseudo labels.

II. RELATED WORK

A. Holistic Methods

Holistic methods directly estimate the object pose and can be divided into two main groups (classical and learning-based methods). PPF [17] uses the point pair global shape feature to retrieve poses from the scene point cloud. Hinterstoisser [18] proposes a new feature which uses RGB colour gradient and 3D surface normal information. These classical methods are not robust in clustered scenes. Based on deep neural networks, some methods transform pose estimation problems into regression problems. DenseFusion [19] uses dense pixel-level fusion to fuse RGB and point cloud. Some other works [3], [20], [21], [1], [2], [22] directly regress the position and rotation of an object from a point cloud or depth map. Because the rotation space is nonlinear, direct regression of rotation is challenging.

B. Keypoint-based methods

Keypoint-based methods detect keypoints in the camera coordinate system and establish correspondence between them and keypoints in canonical object-frame coordinate systems. In terms of keypoint selection, BB8 [7] and YOLO-6D [8] predict the projection of the 3D bounding box on the 2D plane. The corners of the bounding box are generally far from the surface of the object and are less representative. Therefore, PVNet [4] and PVN3D [5] use the farthest point sampling algorithm to select keypoints to reduce position errors. For the symmetry objects, the points selected in this way are highly similar and difficult to distinguish. FFB6D [23] uses SIFT [24] to extract features from different angles of 3D models and associate them with the corresponding 3D positions. However, this work [23], [25] is not suitable for texture-less objects. ParametricNet [9] predicts keypoints in shape templates which has two disadvantages. First, chirality problems [9], [26] occur when predicting keypoints of mirror symmetry objects. Second, the selection of keypoints according to object shape parameters is tedious and less robust. These methods fail to consider the geometric similarity of different components of symmetry objects, thus ignoring the pose ambiguity caused by object symmetries. It is worth noting that there is a lack of a robust strategy to select keypoints from symmetry objects in the keypoint prediction stage.

C. Domain Adaptation for 6D Pose Estimation

Sim-to-real transfer is crucial in 6D pose estimation as it bridges the domain gap between synthetic and real observations through robotic visual systems. Some works from domain randomization aim at sampling a wide variety of simulation settings to learn domain-invariant attributes, such as random backgrounds [11], [27] and image augmentations [28]. Some works [12], [29] harness light-weight physically-based render (PBR) data to simulate realistic texture to reduce the gap between the synthetic and real domains. Some works learn a mapping between different visual domains based on generative adversarial networks [13], [30] or other means of feature [14] mapping. However, these methods either depend on a customized simulation strategy to generate high-quality synthetic data or require a specialized network architecture to extract domain-invariant signals. Alternatively, some other methods [10], [31], [15] conduct render-and-compare strategy to self-supervise on unlabeled

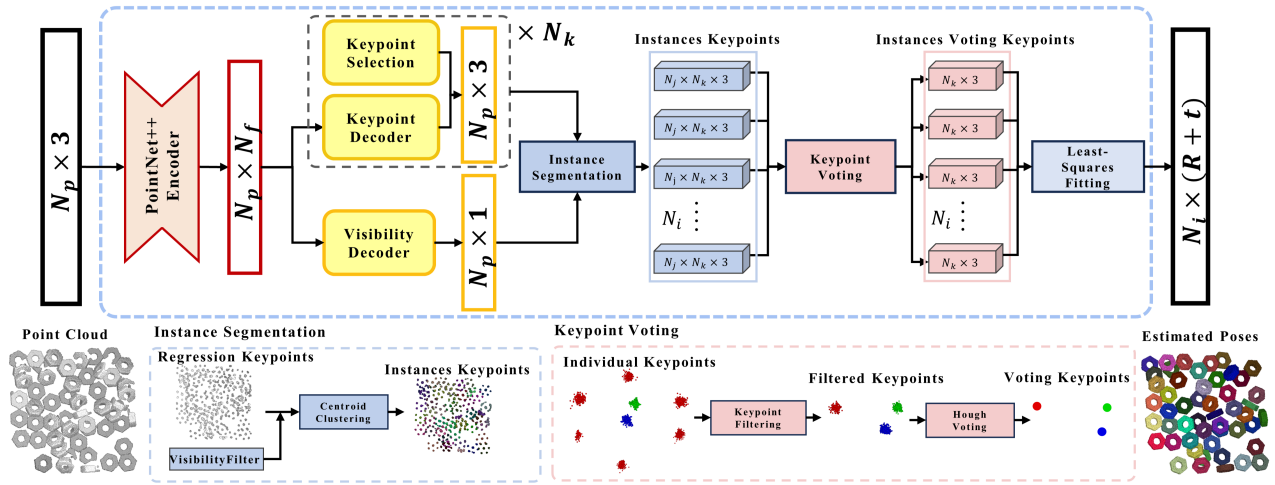


Fig. 2. Overview of SD-Net architecture. SD-Net is constructed based on point-wise keypoint regression and deep Hough voting. It consists of two main parts: symmetric-aware keypoint prediction and self-supervised domain adaptation. Keypoint prediction consists of a novel keypoint selection and filtering algorithm. We omit the domain adaptation framework here for brevity and more details can be found in Section III-C. N_j represents the number of point cloud points for each instance. N_k represents the number of keypoint decoders and corresponds to the number of keypoint of objects. N_i represents the number of instances in the scene.

real data and impose consistencies between rendered features and sensed features. These methods harness 2D appearance information of objects to distinguish reliable predictions. However, texture-less objects provide essentially no 2D effective features, posing a challenge for these methods. In addition, these methods usually focus on pose estimation with domain adaptation of individual objects, ignoring the entire scene.

III. Method

Given a point cloud of a bin-picking scene where multiple object instances are stacked randomly into a pile, we are interested in detecting instances and estimating their rotation $R \in SO(3)$ and translation $t \in \mathbb{R}^3$ in three-dimensional (3D) space. The object pose is represented by a rigid transformation from the object coordinate system to a reference camera coordinate system. In this section, we first introduce our symmetric-aware keypoint selection and filtering algorithm for SD-Net. Then, we present the overall architecture of SD-Net. To boost the performance of object pose estimation in the real world, we introduce the self-supervised domain adaptation framework.

A. Symmetric-Aware keypoint prediction

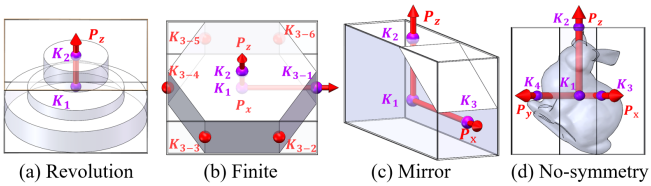
Keypoint Selection. The selection of keypoints is a significant challenge. Keypoints away from the surface of the object will increase localization errors [4]. It is difficult to distinguish points on a symmetry object, which means that the geometric features of different components of the object are similar. To facilitate network convergence, the keypoint selection strategy should meet the following requirements: (1) the selected keypoints should be close to the surface of the object, and (2) the object symmetry should be taken into account. Based on this observation, we propose a heuristic keypoint selection algorithm. A keypoint set is a collection of an object centroid and the points where the object bounding box intersects with the object coordinate axes. In this way, these selected keypoints are close to the surface of the object, making point-based networks easy to aggregate scene context in the vicinity of them. The keypoint set is defined as:

$$K = P_c \cup \{P_i | P_i = P_{boundingbox} \cap P_{axles}\} \tag{1}$$

where P_c represents the centroid, $P_{boundingbox}$ represents the object bounding box and P_{axes} represents the object selected coordinate axes. For example, in Fig. 3(a), the keypoint set contains elements $\{K_1, K_2\}$. Subsequently, an adaptive coordinate axes selection strategy considering the object symmetry class is designed and the details are as follows:

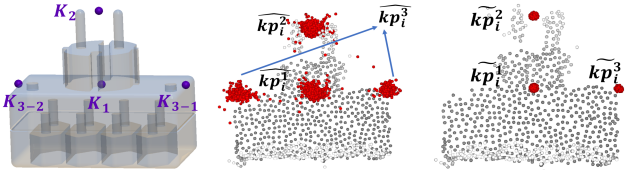
- For revolution symmetry objects, we choose the axis of rotation as P_{axes}. In this way, it can avoid selecting points located on the curved surface which have a low discrimination.
- For finite non-trivial symmetry objects, we choose the axis of rotation and another axis which is randomly selected from the remaining axes as P_{axes}.
- For mirror symmetry objects, we choose two axes which parallel to the plane of mirror symmetry as P_{axes}. This can avoid chirality issues, which is that two point sets of different chirality structures cannot be registered by rotation and translation transformation.
- For no proper symmetry objects, we select all three axes as P_{axes}. This can increase the number of keypoints and improve pose prediction accuracy and robustness.

In addition, for revolution and finite symmetric objects, we transform the object model so that any coordinate axis of the objects coincides with the rotation axis. For mirror symmetry objects, we transform the object model so that any two coordinate axes of the objects coincide with the symmetry plane. Each object instance can then be classified into one of the four symmetry classes above and this keypoint selection can be used for all types of objects.



(a) Revolution (b) Finite (c) Mirror (d) No-symmetry

Fig. 3. The axes selection strategy depends on the object symmetry class. The red axes represent the selected axes. The blue dots represent the selected keypoints, while the red dots represent the corresponding equivalent keypoints.



(a) Keypoint selection (b) Keypoint regression (c) Keypoint filtering

Fig. 4. (a) Keypoints sampled by our selection algorithms on t-less20 object from Siléane dataset. (b) The white points represent the scene instance point clouds, and the red points represent the point-wise predicted keypoints. (c) The red points represent the predicted keypoints after filtering.

Equivalent Keypoint Set. In the process of selecting keypoints using Equation (1), some keypoints are often accompanied by multiple equivalent keypoints, especially for finite symmetric objects. For example, in Fig. 3(b), the keypoint K_3 formed by the intersection of the X-axis and the bounding box contains six equivalent keypoints $\{K_{3-1}, K_{3-3}, ..., K_{3-6}\}$. These keypoints remain geometrically consistent, which can cause ambiguity during the training phase of the network. Therefore, we predict all equivalent keypoints to avoid confusion during the learning phase. Equivalent keypoint set is defined as:

$$K_{j-equivalent} = \{K_j g | g \in G\} \tag{2}$$

where $G \in SO(3)$ is the set of rotation matrix that keeps the object state unchanged. For no proper symmetry objects, G is the unit matrix. K_j is an element in set K in Equation (1).

Keypoint Filtering In the model inference stage (as described in Section III-B), some predicted keypoints will be distributed into multiple candidate clusters around the equivalent keypoint labels, so as the predicted $\widehat{kp_i^3}$ in Fig. 4(b). Moreover, some prediction keypoints have significant deviations from ground truth.

Given the predicted keypoints, to eliminate multiple ambiguities and outlier keypoint candidates, we introduce a robust keypoint filtering algorithm. It is detailed in Algorithm 1. The core step of our algorithm is to cluster the predicted keypoints to form multiple keypoint cluster and find the cluster with the highest point cloud density. To achieve *cluster* in Algorithm 1, we employ DBSCAN [32]. The *density* in Algorithm 1 is used to estimate the density of point cloud clusters and is defined as:

$$D = \frac{1}{N} \sum_{i=1}^{N} \|p_i - p_c\|_2 \tag{3}$$

where p_i and p_c represent the points and centroids in the point cloud clusters, respectively.

As shown in Fig. 4(c), the keypoint filtering algorithm provides a reliable keypoint to fit the pose. For the predicted keypoint $\widehat{kp_i^3}$ distributed into two candidate clusters, the keypoint filtering algorithm dynamically preserves the candidate clusters in the area with the highest density. Meanwhile, the outlier keypoints with significant prediction deviations can be considered as low-density point clouds, so they can be filtered out by the density threshold. It is worth noting that we apply keypoint filtering to each type of keypoint respectively, as different types of keypoints are distributed differently.

Algorithm 1 Density-Based Keypoint Filtering

Input: prediction keypoints $P_1 = \{x_1, x_2, ..., x_n\} \in {}^{n\times 3}$.
Output: filtered keypoints $P_2 = \{x_1, x_2, ..., x_m\} \in {}^{m\times 3}$.

```
initialize an empty cluster set C
C = cluster(P_1)
d_0 = density(C_0)
P_2 = C_0
for C_i in C do
    if density(C_i) < d_0 then
        P_2 ← C_i
        d_0 ← density(C_i)
    end if
end for
return P_2
```

B. Model Architecture

Fig. 2 illustrates our end-to-end 6D pose estimation network. Initially, it takes the point clouds of a bin-picking scene with N_p points as input and applies a feed-forward network PointNet++ [33] for feature extraction. The extracted F_p has a size of $N_p \times N_f$. Subsequently, our network diverges into two decoders which consume F_p to predict the keypoints and visibility for each individual point.

Visibility Decoder. In bin-picking scenarios, some are severely occluded. We are not interested in these instances which cannot be captured at the bottom. Therefore, we set the visibility decoder to reduce the impact of severe occlusion. The point-wise visibility is defined as $V_i = N_i / N_{\max}$. N_i is the number of points of the instance to which the i th point belongs and $N_{\max}$ indicates the highest number of points within all visible instances. We pass F_p into the visibility decoder to predict the point-wise visibility. The visibility loss is defined as:

$$L_v = \frac{1}{N_p} \sum_{i=1}^{N_p} \left\| \widehat{V}_i - V_i \right\| \tag{4}$$

where $\widehat{V}_i$ denotes the prediction of the point-wise visibility.

Keypoint Decoder. Compared with directly predicting the keypoints, predicting the offset of a keypoint relative to the point cloud is more accurate. Therefore, we can pass F_p into a keypoint decoder to predict the point-wise offsets

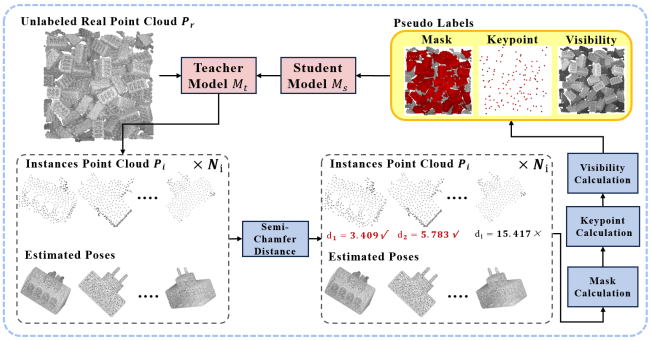


Fig. 5. Overview of our proposed self-supervised domain adaptation framework for 6D object pose estimation. We first train the teacher model on synthetic abundant data to generate initial pose predictions. We then use a 3D geometry pseudo labelling algorithm to distinguish real word predictions for student model training. In the next iteration, the teacher model is initialized as the last trained student model and iterates the above process until the model convergence.

$o_i \in \mathbb{R}^3$. The predicted point-wise keypoint coordinates can be expressed as $\widehat{kp_i} = o_i + p_i$, where p_i denotes the i th point coordinate. Different types of keypoints are predicted separately using independent decoders. For objects with N_k types of keypoints, we use N_k decoders to obtain prediction keypoints $\widehat{kp_i^j} \in \mathbb{R}^3$, which denotes the prediction of the j th keypoint of the instance to which the i th point corresponds. The keypoint prediction loss is defined as:

$$L_k = \frac{1}{N_p} \sum_{j=1}^{N_k} \sum_{i=1}^{N_p} \min_{kp \in kp_i^j} \left\| \widehat{kp_i^j} - kp \right\| \tag{5}$$

where kp_i^j denotes the point-wise keypoint labels ($kp_i^j \in \mathbb{R}^{N_e \times 3}$, N_e denotes the number of equivalent keypoint). This loss function computes the Euclidean distance between the prediction and the closest ground truth.

Generating Pose Hypotheses. In the inference stage, as shown in Fig. 2, we filter out severely occluded scene point clouds and group scene points into instances with Mean Shift algorithm [34] to generate instances keypoints. Then we apply keypoints filtering and hough voting to generate each instance voting keypoints $\{\widehat{K_j} \in \mathbb{R}^3\}_{j=1}^{N_k}$ in the camera coordinate system. When providing their corresponding points $\{K_j\}_{j=1}^{N_k}$ in canonical object coordinate system, we utilize a least-squares fitting algorithm [35] to calculate the optimal values of R and t.

$$L_{lsf} = \sum_{j=1}^{N_k} \left\| \widehat{K_j} - (R \cdot K_j + t) \right\|_2 \tag{6}$$

C. Self-Supervised Domain Adaptation

Fig. 5 summarizes our student-teacher domain adaptation pose estimation framework. We first train a fully-supervised teacher model M_t with abundant labelled synthetic data, as introduced in Section III-B. Then we apply M_t on unlabeled real data to predict object instances poses. Based on these initial poses with decent quality, a robust label selection algorithm is designed to select the best reliable predictions for pseudo labels. The real data with pseudo label generated by pose prediction is utilized to train a student model M_s by self-supervised learning. Crucially, we iterate the above process by taking the M_s as a new teacher model, to progressively boost the quality of pseudo labels and close the domain gap.

Specifically, given unlabeled real point cloud P_r and their initial pose estimations $\{p_i = [R_i|t_i]\}_{i=1}^{N_i}$ generated by the teacher model, we harness geometric constraints to seek the best alignment w.r.t. 6D pose. The core idea is to generate the object point cloud that corresponds to the predicted pose and compare it with the real collected point cloud to determine whether the predicted pose is reliable or not. We conduct the following two steps to leverage geometric constraints. We first backproject the object CAD model C using the corresponding predicted pose to retrieve the point clouds C_i in the camera space: $C_i = R_i \times C + t_i$. Object instances point clouds P_i only contain the surface point cloud that is visible from a particular viewpoint and severely obstructed. P_i is an incomplete point cloud and a subset of C_i. Intuitively, we use the semi-chamfer distance between P_i and C_i as a 3D metric to quantify the quality of the predicted pose:

$$d_i = \frac{1}{N_{P_i}} \sum_{x \in P_i} \min_{y \in C_i} \|x - y\|_2 \tag{7}$$

where x and y denotes 3D points from P_i and C_i respectively, N_{P_i} denotes the number of points for P_i.

We calculate the pose quality for each predicted pose generated by the teacher model M_t and obtain geometry pose quality set $\{d_i\}_{i=1}^{N_i}$ in a bin-picking scene. Then, we dynamically generate a threshold d_g based on the mean and standard deviation of the geometry pose quality set distribution. Pose prediction p_i are regarded as the correct prediction when $d_i < d_g$. The pseudo labels of keypoints for the objects in the real data are calculated according to the correct pose predictions. The pseudo labels of visibility are calculated based on the number of each object instance point cloud. Additionally, for instance, for point clouds whose geometry pose quality $d_i > d_g$, the pseudo labels of the mask are assigned to exclude them from the model training. Specifically, only pseudo labels with a mask label of '1' participate in training. After label generation, we incorporate real point clouds with reliable pseudo labels and train a student model M_s to transfer the knowledge from the synthetic data to real data. In the next iteration, the teacher model M_t is initialized as the last trained student model M_s and iterates the above process until the model convergence.

IV. Experiments

A. Datasets and evaluation metrics

To comprehensively evaluate our method in bin-picking scenarios, we select Siléane dataset [16] and Parametric dataset [9]. Siléane dataset is comprised of a total of more than 2,600 bin-picking scenarios. Parametric dataset consists of two types of data. The L-dataset test set consists of objects with the same parameters as the training set, used to evaluate learning abilities. The G-dataset test set consists of objects

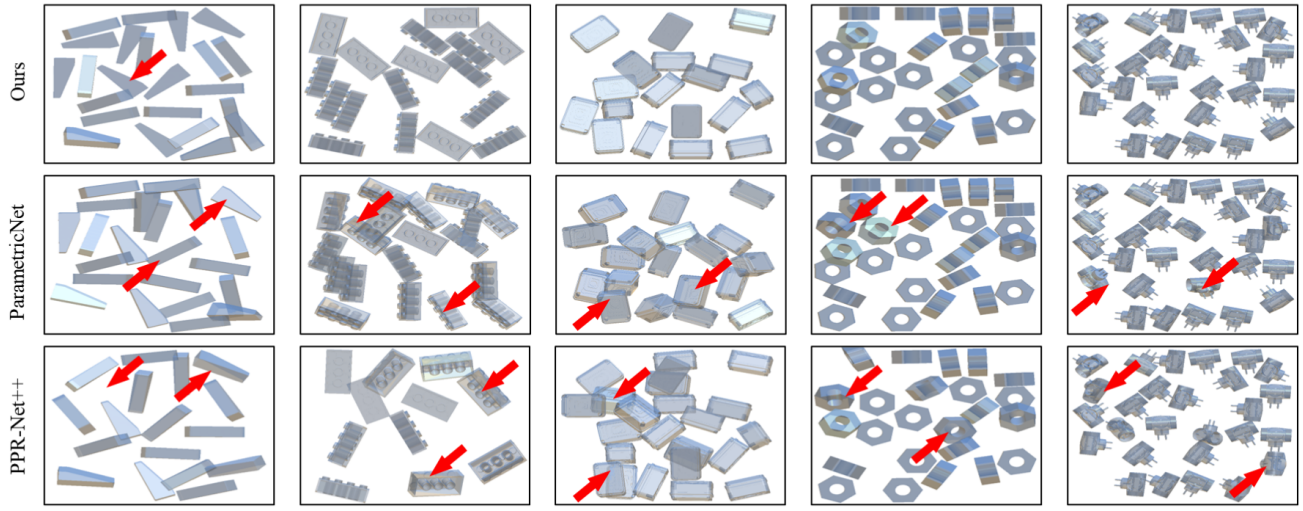


Fig. 6. Qualitative results on Siléane dataset [16]. Rows show a comparison against different methods (SD-Net, ParametricNet and PPR-Net++). Columns show different scenarios. Red arrows highlight the wrong prediction pose and we highlight a maximum of 2 wrong poses in each scene for brevity.

TABLE I

QUANTITATIVE EVALUATION OF 6D POSE ESTIMATION ON SILÉANE DATASET [16]. THE TEST OBJECTS CONTAIN THREE TYPES OF SYMMETRY.

Object Symmetry Class	Bunny Non-Symmetry	C.Stick Revolution	Pepper Revolution	Brick Finite	Gear Revolution	T-Less20 Finite	T-Less22 Non-Symmetry	T-Less29 Finite	Mean
PPF [17]	0.29	0.16	0.06	0.08	0.62	0.20	0.08	0.19	0.21
LINEMOD+ PP[18]	0.45	0.49	0.03	0.39	0.50	0.31	0.21	0.26	0.33
Sock et al [22]	0.74	0.64	0.43	-	-	-	-	-	-
PPR-Net with ICP [3]	0.89	0.95	0.84	-	-	0.85	-	-	-
OP-Net with Lori_1 [1]	0.92	0.94	0.98	0.41	0.82	0.85	0.77	0.51	0.78
OP-Net AP [2]	0.92	0.98	**0.99**	0.45	0.82	0.87	0.84	0.56	0.80
ParametricNet [9]	-	0.97	-	-	**1.00**	0.92	-	0.94	-
PPR-Net++ [20]	0.99	0.98	0.98	0.47	**1.00**	0.93	0.92	0.94	0.90
Ours	**1.00**	**1.00**	**1.00**	**0.75**	**1.00**	**0.98**	**0.96**	**0.98**	**0.96**

with different parameters compared to the training set, used to assess generalization abilities.

We evaluate the performance of methods by calculating the area under the precision-recall curve, which is summarized as the *average precision*. Specifically, when the distance between the predicted pose and ground truth is less than 0.1 times the object's minimum bounding sphere diameter, the prediction pose is considered correct [36].

B. Evaluation on Siléane dataset

Table I summarizes the comparison results between our approach and current 6D pose estimation methods. Our proposed approach achieves state-of-the-art results and outperforms others, obtaining an average precision of 96%. Our approach results in an improvement of +68% to +75% on average precision compared with the conventional approaches [17], [18]. Our approach outperforms the existing deep learning-based methods [1], [2], [9], [20] with a large margin without the need of a separate refinement stage. Fig. 6 shows the qualitative results of SD-Net in severely cluttered bin-picking scenarios and SD-Net is superior to the other two methods in handling symmetry objects.

The observed improvement in finite non-trivial symmetry objects is likely due to the equivalent keypoint set which is easier to learn by the regression network, e.g., T-Less20 and T-Less29 objects. For the brick object, compared with the current state-of-the-art PPR-Net++, SD-Net has a significant increase of 23% in average precision. We observe that smaller object sizes result in larger predicted position and rotation deviations. Our keypoint filtering algorithm eliminates the keypoints with significant prediction deviation, which helps to improve the accuracy of pose estimation.

C. Evaluation on Parametric dataset

Learning ability. From Table II, SD-Net advances state-of-the-art results by 8% on average precision metric. Compared with ParametricNet on TN42 object, SD-Net significantly improves the performance by 26%. A crucial contributing factor is that our keypoints selection algorithm avoids chirality issues (as described in Section III-A). TN06 object has 12 equivalent poses, giving rise to ambiguous pose estimations. SD-Net uses an equivalent keypoint set and the density-based keypoint filtering algorithm to reduce the pose estimation ambiguity and further improve the pose accuracy. In TN16 and TN34 objects scenario, the level of occlusion is relatively low, resulting in highly accurate prediction outcomes for all three methods.

Generalization ability. In Table III, our average precision metric is 8% higher than the state-of-the-art method. SD-Net demonstrates almost the same generalization capability as its

TABLE II

LEARNING ABILITY EVALUATION ON PARAMETRIC DATASET.

Object Symmetry Class	TN06 Finite	TN16 Revolution	TN34 Revolution	TN42 Mirror	Mean
PPR-Net++	0.80	0.99	**1.00**	0.39	0.80
ParametricNet	0.94	**1.00**	**1.00**	0.52	0.87
Ours	**1.00**	**1.00**	**1.00**	**0.78**	**0.95**

TABLE III

GENERALIZATION ABILITY EVALUATION ON PARAMETRIC DATASET.

Train Mode	Method	TN06	TN16	TN34	TN42	Mean
Learn all	PPR-Net++	0.79	0.99	**1.00**	0.28	0.77
	ParametricNet	0.93	**1.00**	**1.00**	0.51	0.86
	Ours	**1.00**	**1.00**	**1.00**	**0.77**	**0.94**
Learn 1/3	PPR-Net++	0.78	0.94	0.96	0.22	0.73
	ParametricNet	0.86	0.98	**1.00**	0.41	0.81
	Ours	**1.00**	**1.00**	**1.00**	**0.67**	**0.92**
Learn 1/5	PPR-Net++	0.77	0.56	0.83	0.18	0.59
	ParametricNet	0.86	0.63	0.86	0.39	0.69
	Ours	**0.98**	**0.65**	**0.89**	**0.54**	**0.77**

learning capability. The results show that SD-Net exhibits strong generalization capability for unseen parameter objects. To further evaluate the SD-Net generalization ability, we conduct experiments by down-sampling the number of training instances to one-third and one-fifth of the original dataset, which means that the model has seen fewer parameter objects during training. Across varying initialization experimental configurations, SD-Net consistently outperforms other methods on generalization. For the TN06 and TN42 objects, SD-Net achieves even higher average precision when trained with only one-fifth of the objects compared to other methods trained on all objects.

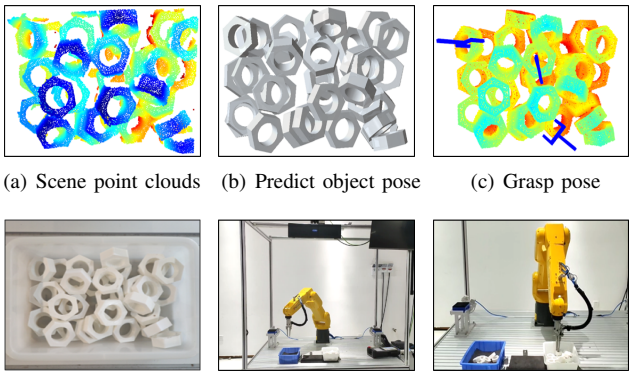
D. Ablation study

We present extensive ablation studies on SD-Net to compare different design choices. Table IV summarises the evaluation results on the Siléane dataset. SD-Net (w/o EKS) learns the keypoint without an equivalent keypoint set. For symmetric objects, its performance is greatly reduced. SD-Net (w/o KF) remove the keypoint filtering algorithm and the performance degradation is large. SD-Net (w/o DA) remove domain adaptation and the results show that our self-training domain adaptation framework enhances the SD-Net performance on real data. SD-Net (w/o SCD) use chamfer distance to quantify the quality of the prediction pose. Due to the incomplete object instances point clouds, chamfer distance cannot precisely quantify the quality. We replace the symmetric-aware keypoint selection strategy with BBox, FPS, and Parameter and do not perform a keypoint filtering algorithm. The performance of these methods is greatly reduced. This shows that our proposed symmetric-aware keypoint prediction performs reliable detection of keypoints. In general, the proposed novel model design can significantly improve average precision.

TABLE IV

ABLATION STUDY ON SILÉANE DATASET. EKS, EQUIVALENT KEYPOINT SET. KF, KEYPOINT FILTERING. SCD, SEMI-CHAMFER DISTANCE. DA, DOMAIN ADAPTATION.

Object	Brick	T-Less20	T-Less22	T-Less29
SD-Net (w/o EKS)	0.08	0.76	0.95	0.66
SD-Net (w/o KF)	0.53	0.03	0.94	0.08
SD-Net (w/o DA)	0.70	0.96	0.95	0.97
SD-Net (w/o SCD)	0.71	0.93	0.94	0.00
SD-Net (BBox)	0.21	0.55	0.61	0.16
SD-Net (FPS)	0.45	0.89	0.88	0.79
SD-Net (Parameter)	0.47	0.92	0.92	0.94
SD-Net	**0.75**	**0.98**	**0.96**	**0.98**



(a) Scene point clouds (b) Predict object pose (c) Grasp pose

(d) Scene RGB image (e) Experiment setup (f) Robotic grasp

Fig. 7. Robotic grasping experiments on the real-world bin-picking scenarios with the proposed approach SD-Net.

E. Real World Experiment

We further explore the application of SD-Net to robot grasping tasks in the real world. We choose the TN06 object instance from Parametric [9] datasets as the grab object. We utilize the Blender platform to generate a simulation data set. Totally 18000 point clouds are annotated, comprising 300 cycles, with each cycle consisting of 60 scenarios. In addition, we collected 600 unlabeled real-world point clouds for self-training domain adaptation.

We deploy the aforementioned well-trained SD-Net on the Fanuc industrial robot, which is equipped with a pneumatic gripper. The whole robot grasping system is implemented using the ROS and MoveIt! frameworks. In the real grasping experiment, objects are stacked in the container, As shown in Fig. 7 (d). The scene point clouds captured by the RVC X 3D camera are cropped, sampled, filtered, and subsequently fed into SD-Net for pose estimation, as shown in Fig. 7 (b). Based on the predicted instance pose, we select the set of grasp configurations from the pre-calculated grasps database, as shown in Fig. 7 (c). We evaluated the ability of SD-Net in 10 grasping trials. Our pipeline can accomplish the robot grasping tasks for all graspable object instances in all trials. It shows that SD-Net demonstrates excellent performance in real robot bin-picking tasks.

V. CONCLUSIONS

In this paper, we propose a new 6D pose estimation network with symmetric-aware keypoint prediction and domain

adaptation. It includes two critical components. We propose a new selection keypoint method which considers object symmetry class and a robust keypoint filtering algorithm. We propose a network-agnostic iterative self-training framework for domain adaptation 6D object pose estimation. Experiments show that SD-Net has significant improvements in average precision compared to state-of-the-art approaches on the public Siléane dataset and Parametric dataset.

REFERENCES

[1] K. Kleeberger and M. F. Huber, "Single shot 6d object pose estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6239–6245.

[2] K. Kleeberger, M. Völk, R. Bormann, and M. F. Huber, "Investigations on output parameterizations of neural networks for single shot 6d object pose estimation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 916–13 922.

[3] Z. Dong, S. Liu, T. Zhou, H. Cheng, L. Zeng, X. Yu, and H. Liu, "PPR-Net: point-wise pose regression network for instance segmentation and 6d pose estimation in bin-picking scenarios," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1773–1780.

[4] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6dof pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4561–4570.

[5] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "PVN3D: A deep point-wise 3d keypoints voting network for 6dof pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 632–11 641.

[6] W. Hua, Z. Zhou, J. Wu, H. Huang, Y. Wang, and R. Xiong, "Rede: End-to-end object 6d pose robust estimation using differentiable outliers elimination," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2886–2893, 2021.

[7] M. Rad and V. Lepetit, "Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3828–3836.

[8] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 292–301.

[9] L. Zeng, W. J. Lv, X. Y. Zhang, and Y. J. Liu, "Parametricnet: 6dof pose estimation network for parametric shapes in stacked scenarios," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 772–778.

[10] G. Wang, F. Manhardt, J. Shao, X. Ji, N. Navab, and F. Tombari, "Self6d: Self-supervised monocular 6d object pose estimation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 108–125.

[11] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1521–1529.

[12] T. Hodaň, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas, "Bop challenge 2020 on 6d object localization," in *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 577–594.

[13] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 35–51.

[14] M. Rad, M. Oberweger, and V. Lepetit, "Domain transfer for 3d pose estimation from color images without manual annotations," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 69–84.

[15] H. Chen, F. Manhardt, N. Navab, and B. Busam, "Texpose: Neural texture learning for self-supervised 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4841–4852.

[16] R. Brégier, F. Devernay, L. Leyrit, and J. L. Crowley, "Symmetry aware evaluation of 3d object detection and pose estimation in scenes of many parts in bulk," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2209–2218.

[17] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *2010 IEEE computer society conference on computer vision and pattern recognition*. Ieee, 2010, pp. 998–1005.

[18] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I 11*. Springer, 2013, pp. 548–562.

[19] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3343–3352.

[20] L. Zeng, W. J. Lv, Z. K. Dong, and Y. J. Liu, "PPR-Net++: accurate 6-d pose estimation in stacked scenarios," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3139–3151, 2021.

[21] X. Zhang, W. Lv, and L. Zeng, "A 6DoF pose estimation dataset and network for multiple parametric shapes in stacked scenarios," *Machines*, vol. 9, no. 12, p. 321, 2021.

[22] J. Sock, K. I. Kim, C. Sahin, and T.-K. Kim, "Multi-task deep networks for depth-based 6d object pose and joint registration in crowd scenarios," *arXiv preprint arXiv:1806.03891*, 2018.

[23] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "Ffb6d: A full flow bidirectional fusion network for 6d pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3003–3013.

[24] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[25] H. Liu, G. Liu, Y. Zhang, L. Lei, H. Xie, Y. Li, and S. Sun, "A 3d keypoints voting network for 6dof pose estimation in indoor scene," *Machines*, vol. 9, no. 10, p. 230, 2021.

[26] L. Zeng, Z.-k. Dong, J.-y. Yu, J. Hong, and H.-y. Wang, "Sketch-based retrieval and instantiation of parametric parts," *Computer-Aided Design*, vol. 113, pp. 82–95, 2019.

[27] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3d orientation learning for 6d object detection from rgb images," in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 699–715.

[28] S. Zakharov, W. Kehl, and S. Ilic, "Deceptionnet: Network-driven domain randomization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 532–541.

[29] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv preprint arXiv:1809.10790*, 2018.

[30] L. Zhao, M. Sun, W. J. Lv, X. Y. Zhang, and L. Zeng, "Domain adaptation on point clouds for 6d pose estimation in bin-picking scenarios," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 2925–2931.

[31] F. Manhardt, G. Wang, B. Busam, M. Nickel, S. Meier, L. Minciullo, X. Ji, and N. Navab, "Cps++: Improving class-level 6d pose and shape estimation from monocular images with self-supervised learning," *arXiv preprint arXiv:2003.05848*, 2020.

[32] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[33] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[34] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

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

[36] R. Brégier, F. Devernay, L. Leyrit, and J. L. Crowley, "Defining the pose of any 3d rigid object and an associated distance," *International Journal of Computer Vision*, vol. 126, no. 6, pp. 571–596, 2018.