BiCo-Net: Regress Globally, Match Locally for Robust 6D Pose Estimation

Zelin Xu[1] , **Yichen Zhang**[1] , **Ke Chen**[1,2,*] and **Kui Jia**[1,2,*]
[1]South China University of Technology
[2]Peng Cheng Laboratory
{eexuzelin, eezyc}@mail.scut.edu.cn, {chenk, kuijia}@scut.edu.cn

Abstract

The challenges of learning a robust 6D pose function lie in 1) severe occlusion and 2) systematic noises in depth images. Inspired by the success of point-pair features, the goal of this paper is to recover the 6D pose of an object instance segmented from RGB-D images by locally matching pairs of oriented points between the model and camera space. To this end, we propose a novel Bi-directional Correspondence Mapping Network (BiCo-Net) to first generate point clouds guided by a typical pose regression, which can thus incorporate pose-sensitive information to optimize generation of local coordinates and their normal vectors. As pose predictions via geometric computation only rely on one single pair of local oriented points, our BiCo-Net can achieve robustness against sparse and occluded point clouds. An ensemble of redundant pose predictions from locally matching and direct pose regression further refines final pose output against noisy observations. Experimental results on three popularly benchmarking datasets can verify that our method can achieve state-of-the-art performance, especially for the more challenging severe occluded scenes. Source codes are available at https://github.com/Gorilla-Lab-SCUT/BiCo-Net.

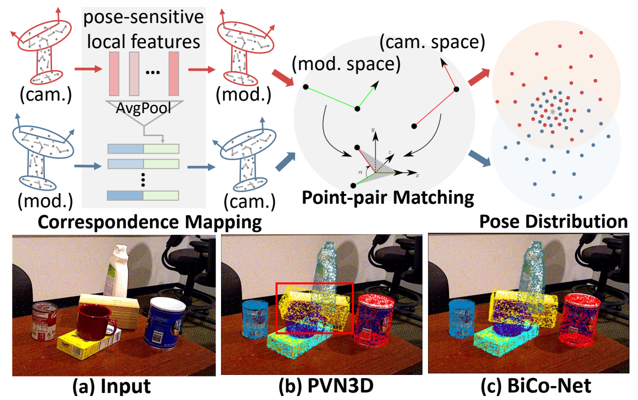

Figure 1: Top: Visualization of our BiCo-Net consisting of deep correspondence mapping, pose computation via point pair matching, and an ensemble of pose predictions. Bottom: Comparative results of existing PVN3D and our BiCo-Net with an example from the YCB-Video benchmark, where the red rectangle highlights a failure of 6D pose estimation by the PVN3D due to inter-object occlusion.

1 Introduction

The problem of 6 Degree-of-Freedom pose estimation aims to predict the orientation and location of one detected object instance in 3D space from a canonical model via recovering a rigid transformation from the object space to the camera space. Such problem has been widely encountered in fields of engineering such as robotic grasping and autonomous driving. A large number of deep methods including regression based [Xiang *et al.*, 2018; Park *et al.*, 2019] and keypoint-based [Tekin *et al.*, 2018; Peng *et al.*, 2019] only rely on extracting texture information from RGB images, which are sensitive to objects with poor textures. Alternatively, with the development and wide application of depth sensors, exploring 6D pose estimation on RGB-D images becomes popular in recent years, where depth images can provide complementary geometry information to RGB images.

The pioneering works [Xiang *et al.*, 2018; Li *et al.*, 2018] on RGB-D images are in a two-stage structure: deep pose estimation and post-processing refinement with iterative closest point (ICP), which leads to less efficiency during inference. Residual learning based refinement in [Wang *et al.*, 2019a] is proposed to improve efficiency with orders of magnitude faster than the ICP, which can ensure real-time inference. From a practical perspective, there remain two compounded challenges for learning a robust 6D pose function to address the problem: 1) severe occlusion and 2) unavoidable systematic noises during depth imaging [Barron and Malik, 2013].

Most of the existing methods [Wang *et al.*, 2019a; He *et al.*, 2021] concern on coping with the former by improving feature encoding on integrating textural and shape features into discriminative representation, while very few work pays attention to the latter. In [Zhou *et al.*, 2021], a point refinement network is the first attempt to explicitly polish point clouds via completion and denoising, whose features are combined with those encoded from raw point clouds and RGB images for better multi-modal feature fusion to regress final poses. However, performance gain of PR-GCN in [Zhou *et al.*, 2021]

*Corresponding authors

over the pose regression baseline can be sensitive to point cloud generation, which itself remains an active and challenging task especially under incomplete and noisy observations.

We argue that *encoding pose-sensitive local features* and *modeling a statistical distribution of pose inliers* are two key factors for accurate and robust performance in 6D pose estimation. On one hand, pose estimation dependent on local texture and geometry can perform stably when missing a part of object regions, which is thus robust against occlusion, but those local features are sensitive to the quality (*e.g.* noises and resolution) of the acquired data. On the other hand, the distribution of pose predictions on local features can be explored to alleviate negative effects of systematic noises in depth imaging. In this paper, we propose a novel deep model for 6D pose estimation on RGB-D images, *i.e.* a Bi-directional Correspondence Mapping Network (BiCo-Net) as shown in Figure 1, by simultaneously addressing both challenges in a unified and implicit manner.

Given the point cloud (generated from the depth image) and the RGB image of an object under the observation pose, the feature output of DenseFusion feature encoder [Wang *et al.*, 2019a] is decoded to generate the corresponding oriented points in the canonical space. To effectively exploit object model priors as a reference, a clean and complete model point cloud under the observed pose is similarly produced in an encoder-decoder learning style from the corresponding oriented points under the canonical pose, which is in an opposite direction of the aforementioned point cloud generation. Inspired by the success of point-pair features [Drost *et al.*, 2010], a set of oriented point pairs from the input (*e.g.* the camera space) can be randomly sampled to produce accurate pose predictions by matching with those corresponding ones in the output space (*e.g.* the model space). As pose prediction relies only on one pair of local oriented points, such a characteristic can encourage robustness against sparse and occluded point clouds. Moreover, owing to revealing a global distribution of rigid pose transformation favored by multiple point pairs, selecting and combining pose predictions of the bi-directional correspondence mapping can alleviate negative effects of outliers in noisy point clouds. For imposing pose-sensitive information, features of the bi-directional point cloud generation can be regularized by a typical pose regression as [Wang *et al.*, 2019a].

In general, the whole network consists of an ensemble of two parts: direct pose regression on a concatenation of global features and point-wise features as [Wang *et al.*, 2019a] and pose computation via locally matching of oriented point pairs in-between canonical and observed poses, which can further refine pose predictions. Extensive experiments on three popular benchmarks, *i.e.* YCB-Video [Xiang *et al.*, 2018], LineMOD [Hinterstoisser *et al.*, 2011] and Occlusion LineMOD [Brachmann *et al.*, 2014], can verify superior performance of the proposed BiCo-Net to the state-of-the-art methods, especially for severe occluded scenes. Main contributions of this paper lie as follows:

- This paper proposes a novel 6D pose estimation method – the BiCo-Net based on locally matching oriented point pairs between the model and camera space and direct pose regression.
- The proposed BiCo-Net is implicitly robust against occlusion and sparse point distribution owing to exploiting the pose-sensitive characteristic of each single pair of oriented points under different poses.
- Negative effects of outliers in noisy depth images can be mitigated via selection and an ensemble of redundant pose predictions.
- Our BiCo-Net can gain state-of-the-art performance on three benchmarks, especially on the more challenging Occlusion LineMOD dataset.

2 Related Works

Keypoint-based 6D Pose Estimation. A typical keypoint-based pose estimation algorithm is designed in a two-stage pipeline: first localizing 2D projection of pre-defined key points in 3D space, and pose predictions can be generated via 2D-to-3D key point correspondence with a PnP [Lepetit *et al.*, 2008]. Existing methods can be categorized into two groups: object detection based [Rad and Lepetit, 2017; Tekin *et al.*, 2018] and dense heatmap based [Oberweger *et al.*, 2018; Pavlakos *et al.*, 2017]. The former performs well on localizing sparse key points of object foreground, but are sensitive to occlusion [Oberweger *et al.*, 2018]. The latter group of methods are more robust to inter-object occlusion in a dense correspondence mapping style. PVNet [Peng *et al.*, 2019] is proposed to detect 2D keypoints via voting on pixel-wise predictions of the directional vector that points to keypoints and is robust to truncation and occlusion. PVN3D [He *et al.*, 2020] extends the 2D keypoints into 3D space by building 3D-3D correspondence and then uses the least-squares fitting [Arun *et al.*, 1987] to generate the pose prediction. Learning of directional vectors pointing to 3D keypoints under camera space suffers from the variation of object pose and the vast 3D search space. In contrast, the bi-directional correspondence mapping in our method makes point-wise predictions on 3D oriented points between the model and camera space directly to build up dense correspondence, instead of their projection in 2D images or the directional vector pointing to 3D keypoints, which enhances local feature discrimination via direct regression based regularization in a pose-sensitive manner.

Dense Regression-based 6D Pose Estimation. An alternative group of algorithms to cope with occlusion is to produce dense pose predictions for each pixel or local patches with hand-crafted features [Liebelt *et al.*, 2008; Sun *et al.*, 2010], CNN patch-based feature encoding [Doumanoglou *et al.*, 2016; Kehl *et al.*, 2016] and CNN pixel-based feature encoding [Wang *et al.*, 2019a; Zhou *et al.*, 2021], whose final pose output is selected via a voting scheme. DenseFusion [Wang *et al.*, 2019a] fuses RGB and Depth information for each point and uses point-wise features to regress dense poses. Zhou *et al.* proposes PR-GCN [Zhou *et al.*, 2021] to handle incomplete and noisy point clouds in practice via designing a PRN to complete and denoise observed point clouds and use graph convolution to better integrated the RGB and Depth information. In [Wang *et al.*, 2019b], normalized object coordinate space is proposed to build up 3D-3D correspondences for each pixel in category level then recover the

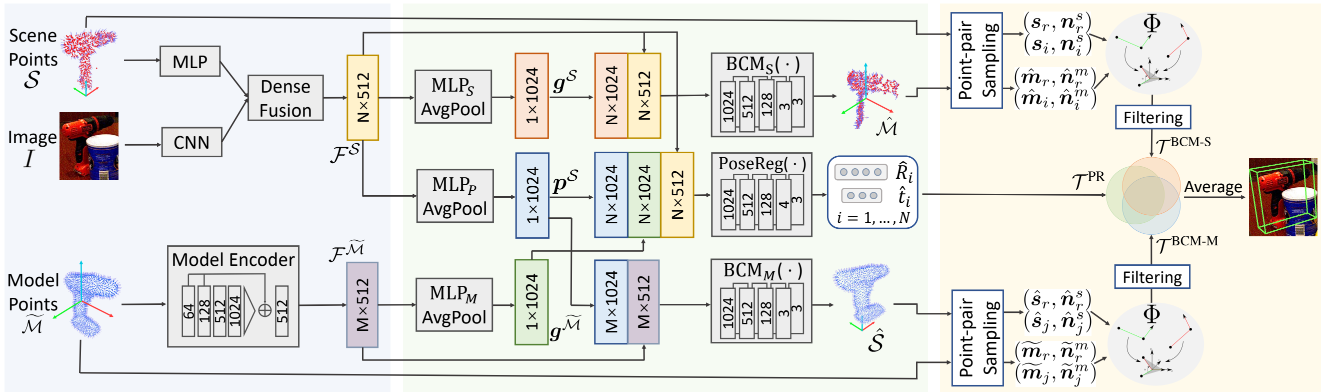

Figure 2: Pipeline of the proposed BiCo-Net.

6D pose and size by the least-squares fitting. The proposed BiCo-Net is designed to both regress dense poses from point-wise features and locally match pose-sensitive oriented points in a unified framework. The complementary characteristics of dense pose regression and dense correspondence mapping can be fully utilized to gain robust pose predictions. Moreover, compared with the least-squares fitting, our point-pair pose computation which uses the correspondence of two oriented points can perform more robust under heavy occlusion.

3 Methodology

The problem of 6D pose estimation of an object given the RGB-D image and its canonical CAD model is to estimate their rotation $\boldsymbol{R} \in SO(3)$ and translation $\boldsymbol{t} \in \mathbb{R}^3$, which can be defined as a rigid transformation $\boldsymbol{T} = [\boldsymbol{R}|\boldsymbol{t}]$ from the model space with respect to the camera space. The whole pipeline of our BiCo-Net is illustrated in Figure 2, which can be divided into three modules: feature encoding on segmented object instances (see Sec. 3.1) in a blue block, generation of oriented points via Bi-directional Correspondence Mapping (see Sec. 3.2) in a green block, and an ensemble of pose predictions by point pair matching and point-wise regression (see Sec. 3.3) in a yellow block.

3.1 Instance Segmentation and Feature Encoding

Following existing methods [Wang *et al.*, 2019a; Zhou *et al.*, 2021], BiCo-Net first employs an off-the-shelf instance segmentation method for RGB images (*e.g.* Mask RCNN [He *et al.*, 2017] in our experiments) to segment the object of interests, which produces a cropped image patch I and one scene point cloud under the camera space $\mathcal{S} = \{(\boldsymbol{s}_i, \boldsymbol{n}_i^s) \in \mathbb{R}^6\}_{i=1}^N$, where $\boldsymbol{s}_i \in \mathbb{R}^3$ denotes 3D coordinates converted from the masked depth region and $\boldsymbol{n}_i^s \in \mathbb{R}^3$ denotes its normal vectors physically computed by PCA. Both I and $\mathcal{S}$ are fed into CNN-based and MLP-based feature encoders respectively to extract texture and geometric features from heterogeneous data sources, which are further fused by the DenseFusion module introduced in [Wang *et al.*, 2019a] to obtain point-wise features $\mathcal{F}^{\mathcal{S}} = \{\boldsymbol{f}_i^s \in \mathbb{R}^{512}\}_{i=1}^N$. Similarly, for exploiting the model priors, we randomly sample one clean model point cloud $\widetilde{\boldsymbol{m}}$ from the canonical CAD model and compute point-wise normal vectors $\widetilde{\boldsymbol{n}}^m$ to form $\widetilde{\mathcal{M}} = \{(\widetilde{\boldsymbol{m}}_j, \widetilde{\boldsymbol{n}}_j^m) \in \mathbb{R}^6\}_{j=1}^M$ as input of an MLP-based feature encoder to generate point-wise features $\mathcal{F}^{\widetilde{\mathcal{M}}} = \{\widetilde{\boldsymbol{f}}_j^m \in \mathbb{R}^{512}\}_{j=1}^M$.

3.2 Bi-directional Correspondence Mapping

Given features $\mathcal{F}^{\mathcal{S}}$ from visual observation I and $\mathcal{S}$, the BCM-scene (BCM-S) in the top row of Figure 2 aims to regress the corresponding oriented point cloud under the model space $\mathcal{M} = \{(\boldsymbol{m}_i, \boldsymbol{n}_i^m) \in \mathbb{R}^6\}_{i=1}^N$, where $\boldsymbol{m}_i = \boldsymbol{R}^{-1}(\boldsymbol{s}_i - \boldsymbol{t})$ and $\boldsymbol{n}_i^m = \boldsymbol{R}^{-1}\boldsymbol{n}_i^s$. With the point-wise features $\mathcal{F}^{\mathcal{S}}$ as input, we employ a MLP with {512,1024} neurons and an average pooling (AvgPool) to generate a global feature $\boldsymbol{g}^{\mathcal{S}} = \mathtt{AvgPool}(\mathtt{MLP}_S(\mathcal{F}^{\mathcal{S}})) \in \mathbb{R}^{1024}$. Finally, $\hat{\mathcal{M}} = \mathtt{BCM}_S(\mathcal{F}^{\mathcal{S}}, \boldsymbol{g}^{\mathcal{S}})$ for superior robustness to only using $\mathcal{F}^{\mathcal{S}}$, where generated points $\hat{\mathcal{M}} = \{(\hat{\boldsymbol{m}}_i, \hat{\boldsymbol{n}}_i^m)\}_{i=1}^N$. Similarly, the BCM-model (BCM-M) in the bottom row of Figure 2 aims to reconstruct a clean point cloud under the camera space $\widetilde{\mathcal{S}} = \{(\widetilde{\boldsymbol{s}}_j, \widetilde{\boldsymbol{n}}_j^s) \in \mathbb{R}^6\}_{j=1}^M$ from the features under the model space $\mathcal{F}^{\widetilde{\mathcal{M}}}$, where $\widetilde{\boldsymbol{s}}_j = \boldsymbol{R}\widetilde{\boldsymbol{m}}_j + \boldsymbol{t}$ and $\widetilde{\boldsymbol{n}}_j^s = \boldsymbol{R}\widetilde{\boldsymbol{n}}_j^m$. To this end, a global feature $\boldsymbol{p}^{\mathcal{S}} \in \mathbb{R}^{1024}$ aggregated from $\mathcal{F}^{\mathcal{S}}$ together with $\mathcal{F}^{\widetilde{\mathcal{M}}}$ are learning to regress $\widetilde{\mathcal{S}}$, while $\hat{\mathcal{S}} = \mathtt{BCM}_M(\mathcal{F}^{\widetilde{\mathcal{M}}}, \boldsymbol{p}^{\mathcal{S}})$ is the generated point cloud under the camera space by the BCM-M branch. Moreover, to impose pose sensitive information for generation of point clouds in the BCM-S and BCM-M, a direct point-wise pose regression (PR) in the middle row of Figure 2 on $\mathcal{F}^{\mathcal{S}}$, $\boldsymbol{g}^{\widetilde{\mathcal{M}}}$, and $\boldsymbol{p}^{\mathcal{S}}$ to predict $\hat{\mathcal{T}}^{\text{PR}} = \mathtt{PoseReg}(\mathcal{F}^{\mathcal{S}}, \boldsymbol{g}^{\widetilde{\mathcal{M}}}, \boldsymbol{p}^{\mathcal{S}})$, where $\hat{\mathcal{T}}^{\text{PR}} = \{(\hat{\boldsymbol{R}}_i, \hat{\boldsymbol{t}}_i)\}_{i=1}^N$ and $\boldsymbol{g}^{\widetilde{\mathcal{M}}} = \mathtt{AvgPool}(\mathtt{MLP}_M(\mathcal{F}^{\widetilde{\mathcal{M}}}))$.

Loss Functions. We use the Euclidean distance to supervise both BCM branches as follows:

$$L^{\text{BCM-S}} = \frac{1}{N}\sum_i (||\boldsymbol{m}_i - \hat{\boldsymbol{m}}_i|| + \lambda||\boldsymbol{n}_i^m - \hat{\boldsymbol{n}}_i^m||),$$

$$L^{\text{BCM-M}} = \frac{1}{M}\sum_j (||\widetilde{\boldsymbol{s}}_j - \hat{\boldsymbol{s}}_j|| + \lambda||\widetilde{\boldsymbol{n}}_j^s - \hat{\boldsymbol{n}}_j^s||),$$

where λ is a trade-off parameter, $(\boldsymbol{m}_i, \boldsymbol{n}_i^m)$ and $(\widetilde{\boldsymbol{s}}_j, \widetilde{\boldsymbol{n}}_j^s)$ are ground truth oriented points of the BCM-S and BCM-M branches, while $(\hat{\boldsymbol{m}}_i, \hat{\boldsymbol{n}}_i^m)$, $(\hat{\boldsymbol{s}}_j, \hat{\boldsymbol{n}}_j^s)$ are the generated points. To ensure the pose consistency between BCM branches and direct regression branch, we replace the ground truth pose in $(\boldsymbol{m}_i, \boldsymbol{n}_i^m)$, $(\widetilde{\boldsymbol{s}}_j, \widetilde{\boldsymbol{n}}_j^s)$ with the mean of point-wise predicted pose $[\hat{\boldsymbol{R}}_i|\hat{\boldsymbol{t}}_i]$ for symmetric objects. For supervising $\hat{\mathcal{T}}^{\text{PR}}$ with $\mathcal{T} = [\boldsymbol{R}|\boldsymbol{t}]$ in the pose regression branch, we use the ADD Loss [Xiang *et al.*, 2018] for asymmetric objects and ADD-S Loss for symmetric objects:

$$L_i^{\text{PR}} = \begin{cases} \frac{1}{K}\sum_k ||(\boldsymbol{R}\boldsymbol{x}_k + \boldsymbol{t}) - (\hat{\boldsymbol{R}}_i\boldsymbol{x}_k + \hat{\boldsymbol{t}}_i)|| & \text{if asym.}, \\ \frac{1}{K}\sum_k \min_{0<l<K} ||(\boldsymbol{R}\boldsymbol{x}_k + \boldsymbol{t}) - (\hat{\boldsymbol{R}}_i\boldsymbol{x}_l + \hat{\boldsymbol{t}}_i)|| & \text{if sym.}; \end{cases}$$

where K is the number of points sampled from the surface of the CAD model, $[\boldsymbol{R}|\boldsymbol{t}]$ and $[\hat{\boldsymbol{R}}_i|\hat{\boldsymbol{t}}_i]$ are ground truth and point-wise predicted poses respectively. The total loss of our BiCo-Net can thus be written as:

$$L^{\text{Total}} = \frac{1}{N}\sum_i L_i^{\text{PR}} + L^{\text{BCM-S}} + L^{\text{BCM-M}}.$$

3.3 Point Pair Matching and Prediction Ensemble

With generated point clouds $\hat{\mathcal{M}}$ and $\hat{\mathcal{S}}$ under the model and camera space respectively, our goal is to learn a rigid transformation $\boldsymbol{T}$ or $\boldsymbol{T}^{-1}$ between camera and model space. Encouraged by Point-pair feature (PPF) [Drost *et al.*, 2010] to describe object poses by matching local features generated from oriented point pairs, any pair of oriented points in $\mathcal{S}$ and its corresponding ones in $\hat{\mathcal{M}}$ can determine object pose, while similar situation is observed for point pairs between $\widetilde{\mathcal{M}}$ and $\hat{\mathcal{S}}$. Specifically, object pose can be readily obtained by randomly sampling a point-pair $(\boldsymbol{s}_r, \boldsymbol{s}_i)$ from scene points $\mathcal{S}$ or $(\widetilde{\boldsymbol{m}}_r, \widetilde{\boldsymbol{m}}_j)$ from model points $\widetilde{\mathcal{M}}$ and then matching it with the corresponding pair $(\hat{\boldsymbol{m}}_r, \hat{\boldsymbol{m}}_i)$ or $(\hat{\boldsymbol{s}}_r, \hat{\boldsymbol{s}}_j)$ generated by our BCM branches. The transformation from $(\hat{\boldsymbol{m}}_r, \hat{\boldsymbol{m}}_i)$ to $(\boldsymbol{s}_r, \boldsymbol{s}_i)$ is defined as the following [Drost *et al.*, 2010]:

$$\boldsymbol{s}_i = \boldsymbol{T}_{\boldsymbol{s}_r \to x}^{-1} \boldsymbol{R}_x(\alpha) \boldsymbol{T}_{\hat{\boldsymbol{m}}_r \to x} \hat{\boldsymbol{m}}_i,$$

where $\boldsymbol{T}_{\hat{\boldsymbol{m}}_r \to x}$ denotes a transformation that translates $\hat{\boldsymbol{m}}_r$ into the origin and rotates $\hat{\boldsymbol{n}}_r^m$ on to the x-axis, and the same definition of $\boldsymbol{T}_{\boldsymbol{s}_r \to x}$ for transforming $(\boldsymbol{s}_r, \boldsymbol{n}_r^s)$. When $\hat{\boldsymbol{m}}_r$ and $\boldsymbol{s}_r$ are aligned to the x-axis, there is a α angle difference about the x-axis between $\boldsymbol{T}_{\boldsymbol{s}_r \to x}\boldsymbol{s}_i$ and $\boldsymbol{T}_{\hat{\boldsymbol{m}}_r \to x}\hat{\boldsymbol{m}}_i$, which encourages to use $\boldsymbol{R}_x(\alpha)$, a rotation with respect to the x-axis, to align $\boldsymbol{T}_{\boldsymbol{s}_r \to x}\boldsymbol{s}_i$ and $\boldsymbol{T}_{\hat{\boldsymbol{m}}_r \to x}\hat{\boldsymbol{m}}_i$. As a result, the rigid transformation $\Phi : (\boldsymbol{s}_r, \boldsymbol{s}_i, \hat{\boldsymbol{m}}_r, \hat{\boldsymbol{m}}_i) \to \boldsymbol{T}$ can be written as:

$$\Phi(\boldsymbol{s}_r, \boldsymbol{s}_i, \hat{\boldsymbol{m}}_r, \hat{\boldsymbol{m}}_i) = \boldsymbol{T}_{\boldsymbol{s}_r \to x}^{-1} \boldsymbol{R}_x(\alpha) \boldsymbol{T}_{\hat{\boldsymbol{m}}_r \to x}.$$

Note that, 6D object pose by locally matching a pair of oriented points under the camera and model space can be readily computed for desirable real-time inference.

As our method only relies on each single point pair to estimate 6D object pose, it allows sparse and imbalanced point distributions, which is thus able to achieve good performance for severe occlusion. For alleviating noises in point clouds and increasing inference speed, we use the FPS algorithm to downsample $\mathcal{S}$ and $\widetilde{\mathcal{M}}$ to a subset of Z points, which are then to generate Z^2 point pairs to compute pair-wise pose candidates $\boldsymbol{T}_z = [\boldsymbol{R}_z|\boldsymbol{t}_z]$ for the BCM-S and BCM-M respectively. For avoiding unreliable pose candidates from point-pairs constructed by neighboring points, pose predictions will be filtered out with the following error measure:

$$\mathcal{E}(\boldsymbol{T}_z) = \frac{1}{N}\sum_i ||(\boldsymbol{R}_z^{-1}(\boldsymbol{s}_i - \boldsymbol{t}_z)) - \hat{\boldsymbol{m}}_i||,$$

and preserving the top 10% of candidates as the pose prediction sets $\mathcal{T}^{\text{BCM-S}}$ and $\mathcal{T}^{\text{BCM-M}}$ of these two branches.

An Ensemble of Pose Predictions. As mentioned in Sec. 1, we obtain three sets of pose predictions (*i.e.* $\mathcal{T}^{\text{PR}}$ from direct pose regression; $\mathcal{T}^{\text{BCM-S}}$ and $\mathcal{T}^{\text{BCM-M}}$ via locally matching with point pairs), from three branches of the BiCo-Net. For achieving superior robustness via using the complementary information of three sets, we consider applying the average pose of $\mathcal{T}^{\text{PR}} \cup \mathcal{T}^{\text{BCM-S}} \cup \mathcal{T}^{\text{BCM-M}}$ as the final pose output.

4 Experiments

4.1 Datasets and Settings

Datasets. To evaluate our BiCo-Net comprehensively, experiments are conducted on three popular benchmarks – the YCB-Video dataset [Xiang *et al.*, 2018], the LineMOD [Hinterstoisser *et al.*, 2011], and the more challenging Occlusion LineMOD [Brachmann *et al.*, 2014]. The YCB-Video dataset has 92 videos in total, each of which shows a subset of 21 objects with varying textures and sizes under cluttered indoor environment. Following existing works [Wang *et al.*, 2019a; Zhou *et al.*, 2021], we adopt 16,189 frames from 80 videos with an additional 80,000 synthetic images provided by [Xiang *et al.*, 2018] for training and extract 2949 key frames from the remaining 12 videos for testing. The LineMOD contains 15,783 images belonging to 13 low-textural objects placed under different cluttered environments. We use the standard training/testing split as [Xiang *et al.*, 2018; Wang *et al.*, 2019a]. The Occlusion LineMOD provides 6D pose labels of 8 objects selected from the LineMOD and includes 1214 images with multiple heavily occluded objects, which is made more challenging.

Performance Metrics. Following [Wang *et al.*, 2019a; Zhou *et al.*, 2021], we adopt the average distance (ADD) [Xiang *et al.*, 2018] and ADD-Symmetric (ADD-S) as performance metrics. 6D pose predictions are considered to be correct if the ADD/ADD-S is smaller than a predefined threshold. For the YCB-Video dataset, we vary from 0 to 10cm to plot an accuracy-threshold curve and report the area under the curve (AUC). We also report the result of ADD-S $<$2cm as [Wang *et al.*, 2019a; Zhou *et al.*, 2021]. For the LineMOD and the Occlusion LineMOD, we use ADD-S for symmetric objects (*i.e.* eggbox and glue) and ADD for the remaining objects having an asymmetric geometry while taking 10% of the diameter as the threshold.

4.2 Implementation Details

The numbers of scene/model points, *i.e.* , N/M, are set to 1000/1000. In point-pair pose computation, we downsample the scene points and model points to $Z = 100$ points by the

	PoseCNN+ICP		DenseFusion		G2L-Net		PVN3D		PR-GCN		BiCo-Net (ours)	
	AUC	<2cm	AUC	<2cm	AUC	<2cm	AUC	<2cm	AUC	<2cm	AUC	<2cm
002_master_chef_can	95.8	**100.0**	96.4	**100.0**	94.0	-	96.0	**100.0**	**97.1**	**100.0**	96.2	**100.0**
003_cracker_box	92.7	91.6	95.5	99.5	88.7	-	96.1	**100.0**	**97.6**	**100.0**	96.6	**100.0**
004_sugar_box	98.2	**100.0**	97.5	**100.0**	96.0	-	97.4	**100.0**	**98.3**	**100.0**	97.8	**100.0**
005_tomato_soup_can	94.5	96.9	94.6	96.9	86.4	-	**96.2**	**98.1**	95.3	97.6	95.7	**98.1**
006_mustard_bottle	**98.6**	**100.0**	97.2	**100.0**	95.9	-	97.5	**100.0**	97.9	**100.0**	98.0	**100.0**
007_tuna_fish_can	97.1	**100.0**	96.6	**100.0**	84.1	-	96.0	**100.0**	**97.6**	**100.0**	96.5	**100.0**
008_pudding_box	97.9	**100.0**	96.5	**100.0**	93.5	-	97.1	**100.0**	**98.4**	**100.0**	97.5	**100.0**
009_gelatin_box	**98.8**	**100.0**	98.1	**100.0**	96.8	-	97.7	**100.0**	96.2	94.4	**98.8**	**100.0**
010_potted_meat_can	92.7	93.6	91.3	93.1	86.2	-	93.3	94.6	**96.6**	**99.1**	93.0	94.5
011_banana	97.1	99.7	96.6	**100.0**	96.3	-	96.6	**100.0**	**98.5**	**100.0**	97.1	**100.0**
019_pitcher_base	97.8	**100.0**	97.1	**100.0**	91.8	-	97.4	**100.0**	**98.1**	**100.0**	97.6	**100.0**
021_bleach_cleanser	96.9	99.4	95.8	**100.0**	92.0	-	96.0	**100.0**	**97.9**	**100.0**	96.6	**100.0**
024_bowl	81.0	54.9	88.2	**98.8**	86.7	-	90.2	80.5	90.3	96.6	**96.7**	**100.0**
025_mug	95.0	99.8	97.1	**100.0**	95.4	-	97.6	**100.0**	**98.1**	**100.0**	97.0	**100.0**
035_power_drill	**98.2**	99.6	96.0	98.7	95.2	-	96.7	**100.0**	98.1	**100.0**	97.0	99.9
036_wood_block	87.6	80.2	89.7	94.6	86.2	-	90.4	93.8	**96.0**	**100.0**	92.1	90.1
037_scissors	91.7	95.6	95.2	**100.0**	83.8	-	**96.7**	**100.0**	**96.7**	**100.0**	92.2	99.5
040_large_marker	97.2	99.7	97.5	**100.0**	96.8	-	96.7	99.8	**97.9**	**100.0**	97.4	**100.0**
051_large_clamp	75.2	74.9	72.9	79.2	94.4	-	93.6	93.6	87.5	93.3	**94.7**	**98.3**
052_extra_large_clamp	64.4	48.8	69.8	76.3	**92.3**	-	88.4	83.6	79.7	84.6	88.2	**90.2**
061_foam_brick	97.2	**100.0**	92.5	**100.0**	94.7	-	96.8	**100.0**	**97.8**	**100.0**	97.2	**100.0**
ALL	93.0	93.2	93.1	96.8	92.4	-	95.5	97.6	95.8	98.5	**96.0**	**98.8**

Table 1: Comparison of AUC (%) and ADD-S $<$ 2cm (%) ("<2cm" for short) on the YCB-Video dataset. Symmetric objects are highlighted in bold. Comparative methods with the proposed BiCo-Net are PoseCNN+ICP [Xiang *et al.*, 2018], DenseFusion [Wang *et al.*, 2019a], G2L-Net [Chen *et al.*, 2020], PVN3D [He *et al.*, 2020] and PR-GCN [Zhou *et al.*, 2021].

Method	ape	ben.	cam	can	cat	drill.	duck	**egg.**	**glue**	hole.	iron	lamp	phone	MEAN
Implict+ICP [Sundermeyer *et al.*, 2018]	20.6	64.3	63.2	76.1	72.0	41.6	32.4	98.6	96.4	49.9	63.1	91.7	71.0	64.7
SSD6D+ICP [Kehl *et al.*, 2017]	65.0	80.0	78.0	86.0	70.0	73.0	66.0	**100.0**	**100.0**	49.0	78.0	73.0	79.0	79.0
PointFusion [Xu *et al.*, 2018]	70.4	80.7	60.8	61.1	79.1	47.3	63.0	99.9	99.3	71.8	83.2	62.3	78.8	73.7
DenseFusion [Wang *et al.*, 2019a]	79.5	84.2	76.5	86.6	88.8	77.7	76.3	99.9	99.4	79.0	92.1	92.3	88.0	86.2
DenseFusion (Iter.) [Wang *et al.*, 2019a]	92.3	93.2	94.4	93.1	96.5	87.0	92.3	99.8	**100.0**	92.1	97.0	95.3	92.8	94.3
G2L-Net [Chen *et al.*, 2020]	96.8	96.1	98.2	98.0	99.2	**99.8**	97.7	**100.0**	**100.0**	99.0	99.3	99.5	98.9	98.7
PR-GCN [Zhou *et al.*, 2021]	**97.6**	**99.2**	99.4	98.4	98.7	98.8	**98.9**	99.9	**100.0**	**99.4**	98.5	99.2	98.4	98.9
BiCo-Net (Ours)	97.3	98.8	**99.6**	**99.3**	**100.0**	98.9	98.7	99.8	99.8	99.2	**100.0**	**99.7**	**99.2**	**99.3**

Table 2: Comparative evaluation of 6D pose estimation in terms of ADD(-S) (%) on the LineMOD dataset. Objects in bold are symmetric.

Method	PoseCNN	Pix2pose	PVNet	HybridPose	PVN3D	PR-GCN	BiCo-Net
ape	9.6	22.0	15.8	20.9	33.9	40.2	**55.6**
can	45.2	44.7	63.3	75.3	**88.6**	76.2	83.2
cat	0.9	22.7	16.7	24.9	39.1	**57.0**	47.3
drill.	41.4	44.7	65.7	70.2	78.4	**82.3**	69.9
duck	19.6	15.0	25.2	27.9	41.9	30.0	**58.3**
egg.	25.9	25.2	50.2	52.4	**80.9**	68.2	78.1
glue	39.6	32.4	49.6	53.8	68.1	67.0	**76.9**
holep.	22.1	49.5	39.7	54.2	74.7	**97.2**	87.2
MEAN	24.9	32.0	40.8	47.5	63.2	65.0	**69.5**

Table 3: Comparison of ADD(-S) (%) on the Occlusion LineMOD. Symmetric objects are marked in bold. Competing methods with our BiCo-Net are PoseCNN [Xiang *et al.*, 2018], Pix2pose [Park *et al.*, 2019], PVNet [Peng *et al.*, 2019], HybridPose [Song *et al.*, 2020], PVN3D [He *et al.*, 2020] and PR-GCN [Zhou *et al.*, 2021].

FPS which thus generates $Z^2 = 10,000$ pose candidates from point pairs. The hyper-parameter λ in the losses of BCM-S and BCM-M branches is empirically set to 0.05. We use the Adam optimizer with a 10^{-4} learning rate to train our model for 50 epochs, and the learning rate decays 0.3 per 10 epochs.

4.3 Comparison with State-of-the-art Methods

Comparative evaluation of the proposed BiCo-Net and state-of-the-art methods on the YCB-Video, LineMOD and Occlusion LineMOD datasets are showed in Tables 1, 2, and 3. In general, our method can consistently achieve state-of-the-art performance in all benchmarks. Specifically, on the YCB-Video dataset, our method achieves the best performance on both metrics in Table 1, and similar results on the LineMOD in Table 2 can also be observed. Compared to moderate improvement on the YCB-Video and LineMOD, the proposed BiCo-Net can gain accuracy of 69.5% on the more challenging Occlusion LineMOD, which is significantly superior to the state-of-the-art methods as illustrated in Table 3. Such results can verify the effectiveness of our BiCo-Net for 6D pose estimation on RGB-D images. In addition, we measure the inference time on average of the proposed method: the forward time of BiCo-Net is 16ms; the point-pair pose computation time is 29ms; the segmentation network takes 30ms. As a result, the average time for processing a frame for inference is 75ms with a GTX 1080 Ti GPU, which is comparable to existing methods (*e.g.* 60ms for the DenseFusion and 68ms for the PR-GCN) to meet desirable real-time inference in practical applications.

4.4 Ablation Studies

Robustness against Inter-Object Occlusion. To evaluate the robustness of our method against occlusion, we follow

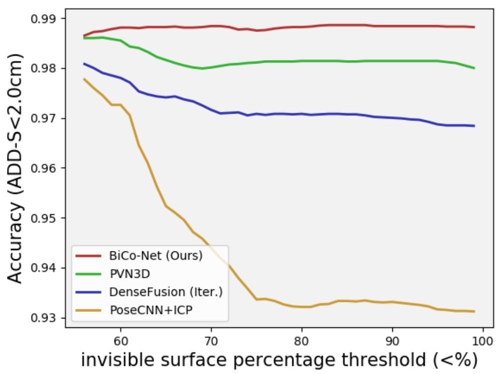

Figure 3: Comparative evaluation with different levels of occlusion on the YCB-Video benchmark.

[Wang *et al.*, 2019a; He *et al.*, 2020] to define occlusion level by using the percentage of invisible points on the object surface. We compare our BiCo-Net with several methods, including PoseCNN+ICP [Xiang *et al.*, 2018], DenseFusion [Wang *et al.*, 2019a], and PVN3D [He *et al.*, 2020], on the YCB-Video with accuracy of ADD-S $<$2cm under different levels of occlusion, whose results are shown in Figure 3. With the percentage of invisible part increasing, our method can perform stably well when performance of comparative methods decrease or even collapse, which again confirms superior robustness of the proposed BiCo-Net to its competitors even with heavily occluded objects.

Pose Reg	BCM-S	BCM-M	Ensemble	YCB-V	LM-O
✓	×	×	×	94.6	62.1
✓	✓	×	×	95.3	63.7
✓	✓	×	✓	95.5	66.9
✓	×	✓	×	95.4	66.1
✓	×	✓	✓	95.5	67.4
✓	✓	✓	×	95.7	67.6
✓	✓	✓	✓	96.0	69.5

Table 4: Effects of BCM branches and an ensemble of pose predictions based on pose regression. We report the AUC (%) on YCB-Video (YCB-V) and ADD(-S) (%) on Occlusion LineMOD (LM-O).

Effects of An Ensemble of Filtered Pose Predictions. To learn a correspondence mapping using multiple instances, a typical robust scheme is least-square fitting based RANSAC to optimize the hypothesis with the maximum inliers. We conduct one experiment in terms of the AUC metrics of the YCB-Video dataset to obtain 79.1%/89.1% on the output of the BCM-S/BCM-M branch, which is significantly inferior to the results using point-pair matching in our BiCo-Net, reaching 94.9%/95.0% only with $\mathcal{T}^{\text{BCM-S}}$/$\mathcal{T}^{\text{BCM-M}}$ (*i.e.* without direct pose regression). Such a result confirms the rationale of ensembling pose predictions in $\mathcal{T}^{\text{BCM-S}}$ and $\mathcal{T}^{\text{BCM-M}}$, owing to the better robustness against outliers in noisy point clouds than the RANSAC. Moreover, we take the average of $\mathcal{T}^{\text{PR}} \cup \mathcal{T}^{\text{BCM-S}} \cup \mathcal{T}^{\text{BCM-M}}$ as the final pose prediction of our BiCo-Net in an ensemble manner, whose effectiveness is verified on the YCB-Video and Occlusion LineMOD respectively (See the last row of Table 4). Moreover, without direct pose regression, as results of $\mathcal{T}^{\text{BCM-S}}$/$\mathcal{T}^{\text{BCM-M}}$ are

Z	2	4	10	20	50	100	200
$\mathcal{T}^{\text{BCM-S}}$	94.2	94.8	95.4	95.5	95.7	95.7	95.7
$\mathcal{T}^{\text{BCM-M}}$	94.5	94.9	95.4	95.6	95.6	95.6	95.6

Table 5: Effects of varying size of points for generating point pairs on the YCB-Video dataset. Z denotes the number of FPS points sampled on input scene points or model points.

only 94.9%/95.0% on the AUC metrics of the YCB-Video dataset, the direct pose regression branch can benefit generation of point clouds in the BCM branches (*i.e.* 95.7%/95.6% for $Z = 100$ in Table 5).

Effects of Bidirectional Correspondence Mapping. For evaluating the effectiveness of the BCM-S and BCM-M branches of our method, we conduct experiments on the YCB-Video dataset and the Occlusion LineMOD. The baseline method takes scene points $\mathcal{S}$ and cropped image patch I as input and only performs point-wise pose regression by $\mathcal{T}^{\text{PR}} = \texttt{PoseReg}(\mathcal{F}^{\mathcal{S}}, \boldsymbol{p}^{\mathcal{S}})$. The average of $\mathcal{T}^{\text{PR}}$ of each object instance is utilized as its pose prediction. As shown in Table 4, the method introducing the BCM-S branch can improve 0.7% and 1.6% on two datasets respectively, indicating that BCM-S effectively improved the discrimination of local feature coding owing to introducing point-wise pose sensitive regularization. The BCM-M branch can outperform the baseline by 0.8% and 4.0% on two datasets respectively. This can be credited to exploiting shape priors of the CAD model provides an ideal reference for the pose regression network, which alleviates the partiality of scene point clouds due to (self-)occlusion. The combination of BCM-S and BCM-M further gains an improvement of 1.1% and 4.5% on both datasets, indicating that these two branches provide complementary information, which further supports our claim about an ensemble of pose predictions.

Evaluation on Size of Point-pairs. We evaluate pose predictions via point-pair matching by taking an average of $\mathcal{T}^{\text{BCM-S}}$ and $\mathcal{T}^{\text{BCM-M}}$ as the output pose respectively with varying size Z of points, and report results in Table 5. As our method only relies on simple geometric properties of two local points for pose computation, even with sparse input points, $\mathcal{T}^{\text{BCM-S}}$/$\mathcal{T}^{\text{BCM-M}}$ can gain comparable performance to $\mathcal{T}^{\text{PR}}$ (see the first row of Table 4).

5 Conclusion

This paper introduces a novel neural network for 6D pose estimation based on an ensemble of direct regression and locally matching pairs of oriented points under the camera and model space. The proposed BiCo-Net can achieve robust performance on severe inter-object occlusion and systematic noises in scene point clouds, owing to our design of exploiting pose sensitive information carried by each pair of oriented points and an ensemble of redundant pose predictions. Experiment results can verify the effectiveness of each module in our method and the state-of-the-art performance.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (Grant No.: 61771201, 61902131), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No.: 2017ZT07X183).

References

[Arun *et al.*, 1987] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *TPAMI*, 1987.

[Barron and Malik, 2013] Jonathan T Barron and Jitendra Malik. Intrinsic scene properties from a single rgb-d image. In *CVPR*, 2013.

[Brachmann *et al.*, 2014] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *ECCV*, 2014.

[Chen *et al.*, 2020] Wei Chen, Xi Jia, Hyung Jin Chang, Jinming Duan, and Ales Leonardis. G2l-net: Global to local network for real-time 6d pose estimation with embedding vector features. In *CVPR*, 2020.

[Doumanoglou *et al.*, 2016] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malassiotis, and Tae-Kyun Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *CVPR*, 2016.

[Drost *et al.*, 2010] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*, 2010.

[He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[He *et al.*, 2020] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *CVPR*, 2020.

[He *et al.*, 2021] Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. *CVPR*, 2021.

[Hinterstoisser *et al.*, 2011] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *ICCV*, 2011.

[Kehl *et al.*, 2016] Wadim Kehl, Fausto Milletari, Federico Tombari, Slobodan Ilic, and Nassir Navab. Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In *ECCV*, 2016.

[Kehl *et al.*, 2017] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *ICCV*, 2017.

[Lepetit *et al.*, 2008] Vincent Lepetit, Francesc Moreno-Noguer, and P. Fua. Epnp: An accurate o(n) solution to the pnp problem. *IJCV*, 2008.

[Li *et al.*, 2018] Chi Li, Jin Bai, and Gregory D Hager. A unified framework for multi-view multi-class object pose estimation. In *ECCV*, 2018.

[Liebelt *et al.*, 2008] Joerg Liebelt, Cordelia Schmid, and Klaus Schertler. independent object class detection using 3d feature maps. In *CVPR*, 2008.

[Oberweger *et al.*, 2018] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *ECCV*, 2018.

[Park *et al.*, 2019] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *ICCV*, 2019.

[Pavlakos *et al.*, 2017] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *ICRA*, 2017.

[Peng *et al.*, 2019] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019.

[Rad and Lepetit, 2017] Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *ICCV*, 2017.

[Song *et al.*, 2020] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *CVPR*, 2020.

[Sun *et al.*, 2010] Min Sun, Gary Bradski, Bing-Xin Xu, and Silvio Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *ECCV*, 2010.

[Sundermeyer *et al.*, 2018] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *ECCV*, 2018.

[Tekin *et al.*, 2018] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *CVPR*, 2018.

[Wang *et al.*, 2019a] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *CVPR*, 2019.

[Wang *et al.*, 2019b] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, 2019.

[Xiang *et al.*, 2018] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *RSS*, 2018.

[Xu *et al.*, 2018] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *CVPR*, 2018.

[Zhou *et al.*, 2021] Guangyuan Zhou, Huiqun Wang, Jiaxin Chen, and Di Huang. Pr-gcn: A deep graph convolutional network with point refinement for 6d pose estimation. In *ICCV*, 2021.