

# DGECN: A Depth-Guided Edge Convolutional Network for End-to-End 6D Pose Estimation

Tuo Cao<sup>1</sup>, Fei Luo<sup>1\*</sup>, Yanping Fu<sup>2</sup>, Wenxiao Zhang<sup>1</sup>, Shengjie Zheng<sup>1</sup>, and Chunxia Xiao<sup>1\*</sup>

<sup>1</sup>School of Computer Science, Wuhan University, Wuhan, Hubei, China

<sup>2</sup>School of Computer Science and Technology, Anhui University, Hefei, Anhui, China

ypfu@ahu.edu.cn, wenxxiao.zhang@gmail.com, zsj\_mdk@163.com, {maplect, luofei, cxxiao}@whu.edu.cn

<http://graphvision.whu.edu.cn/>

## Abstract

Monocular 6D pose estimation is a fundamental task in computer vision. Existing works often adopt a two-stage pipeline by establishing correspondences and utilizing a RANSAC algorithm to calculate 6 degrees-of-freedom (6DoF) pose. Recent works try to integrate differentiable RANSAC algorithms to achieve an end-to-end 6D pose estimation. However, most of them hardly consider the geometric features in 3D space, and ignore the topology cues when performing differentiable RANSAC algorithms. To this end, we proposed a **Depth-Guided Edge Convolutional Network (DGECN)** for 6D pose estimation task. We have made efforts from the following three aspects: 1) We take advantages of estimated depth information to guide both the correspondences-extraction process and the cascaded differentiable RANSAC algorithm with geometric information. 2) We leverage the uncertainty of the estimated depth map to improve accuracy and robustness of the output 6D pose. 3) We propose a differentiable Perspective-n-Point (PnP) algorithm via edge convolution to explore the topology relations between 2D-3D correspondences. Experiments demonstrate that our proposed network outperforms current works on both effectiveness and efficiency.

## 1. Introduction

Object pose estimation is a task of calculating the 6 degrees of freedom (DoF) pose of a rigid object, including its location and orientation in an image. It is widely used in the three-dimensional registration of AR [1, 28, 45], robotic vision [27, 31] and 3D reconstruction [9, 10]. Due to the presence of noises and other influential factors, such as the occlusion, noisy background, and illumination variations, accurately estimating the 6DoF poses of the objects in the RGB image is still a challenging problem.

\*Chunxia Xiao and Fei Luo are co-corresponding authors

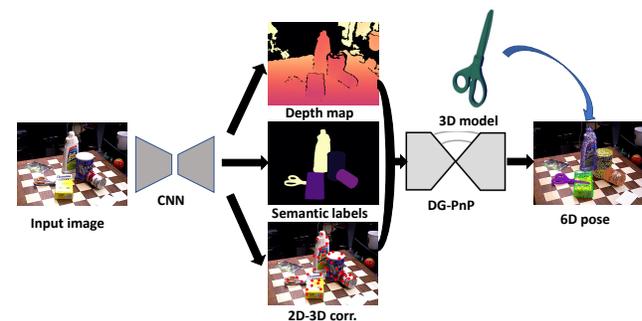


Figure 1. **Pipeline of DGECN.** With an input RGB image, we propose a novel DGECN to simultaneously predict segmentation and depth maps. After established 2D-3D correspondences, we replace the RANSAC/PnP with a learnable DG-PnP to regress 6D pose.

Current object pose estimation methods can be divided into two types: 1) the object poses are estimated using a single RGB image [17, 27, 28, 31, 45] or 2) an RGB image accompanying a depth image [14, 39, 41]. For both RGB based and RGB-D based methods, the keypoints-based works are dominant in this field. On the other hand, methods based on direct regression are usually inferior to keypoints-based methods. The keypoints-based methods usually consist of two stages: firstly it predicts the 2D location of the keypoints of the 3D model on RGB images via a modern neural network. And then calculate the 6D pose parameters with the RANSAC-based Perspective-n-Point (PnP) method from 2D-3D correspondences. Although many representative works [15, 22, 25, 33, 35, 36] have proven the validity of the two-stage pipeline, there are still many limitations in it. Firstly, few methods can directly output the 6D pose parameters. Most of the existing methods still use a variant of the RANSAC-based PnP algorithm to estimate the pose parameters. Secondly, RANSAC-based PnP can be very time-cost when the 2D-3D correspondences are dense. Thirdly, the

network in most two-stage works cannot directly output 6D pose, so their loss functions cannot optimize our expected pose estimation. Finally, the two-stage estimation may lead to significant accumulative error, which gradually increases among the two connected steps.

Recently, some works try to integrate a differentiable RANSAC algorithm into the pipeline, so the network can be trained end-to-end. Brachmann *et al.* [3] proposed a differentiable PnP method. Hu *et al.* [16] leveraged PointNet [29] to approximate PnP for sparse correspondences. But these works either require a cumbersome training process or do not consider the geometry clues. Wang *et al.* [42] made an end-to-end framework by replacing RANSAC-based PnP with Patch-PnP, this method works well, but it relies on the Dense Correspondences Map and Surface Region Attention Map in their network. It can hardly directly learning 6D pose from 2D-3D correspondences.

To this end, we propose Depth-Guided Edge Convolutional Network (DGEEN), jointly handling the correspondences extraction and the 6D pose estimation. Our network leverages a depth guided network to establish 2D-3D correspondences and learn the 6D pose from the correspondences by a novel Dynamic Graph PnP (DG-PnP). On one hand, depth information allows us to make full use of the geometric constraint of rigid objects. On the other hand, we fully revisit the properties of correspondence set and find it can better handle complex textures by constructing a graph structure. Our end-to-end pipeline is shown in Fig. 1.

Experimental results on LM-O [2] and YCB-V [5, 45] demonstrate our network is comparable even superior to the state-of-the-art methods in terms of accuracy and efficiency.

Our contributions in this work can be summarized as follows:

- We propose a Depth-guided network to directly learn the 6D pose from a monocular image without additional information required. Furthermore, we propose a Depth Refinement Network (DRN) to polish the quality of the estimated depth map.
- We explore the properties of 2D correspondence sets and discover that 6D pose parameters can be learned better from the 2D keypoint distributions by constructing a graph. We further propose a simple but effective Dynamic Graph PnP (DG-PnP) to directly learn 6D pose from 2D-3D correspondences.

## 2. Related work

**Direct Methods.** These methods usually directly estimate the 6D pose in a single shot. Some early works leverage template matching techniques. However, they do not perform satisfactorily under occlusion. With the advance of deep learning, some works regress the pose parameters via a network. Xiang *et al.* [45] first introduced CNN into this

field, they employed a network based on GoogleNet [38] to directly learn the 6D camera pose. This problem is still challenging due to the variety of objects as well as the complexity of a scene caused by clutter and occlusions between objects. To address this flaw, PoseCNN [45] estimated the 3D translation of an object by localizing its center in the image and predicting its distance from the camera. However, this problem is still difficult due to the non-closed property to addition of rotation matrix. Some works [49] utilized the  $\mathbb{SO}(3)/\mathbb{SE}(3)$  to make the rotation space differentiable.

**Correspondence-based Methods.** The methods based on 2D-3D correspondence detection have gradually become the mainstream in object pose estimation. PVNet [28] and Seg-Driven [17] conducted segmentation coupled with voting for each correspondence to make the estimation more robust. EPOS [15] made use of surface fragments accounting for ambiguities in pose. Pix2Pose [27] used a network based on GAN to predict the 3D coordinates of each object pixel without textured models. Oberweger *et al.* [26] output pixel-wise heatmaps of keypoints to address the issue of occlusion. Recent years, a few works aim to avoid the time-consuming RANSAC-based PnP in keypoint-based pipeline. DSAC [3] presented two alternative ways to make RANSAC differentiable by soft argmax and probabilistic selection and applied it to the problem of camera localization. Single-Stage [16] employed a PointNet-like architecture to learn the 6D pose from 2D-3D correspondences. However, this method can only deal with the sparse correspondences. To avoid this, GDR-Net [42] let the network predict the surface regions as additional ambiguity-aware supervision and used them within their Patch-PnP framework. SO-Pose [7] focused on the occluded part to encode the geometric features of the object more completely and accurately.

**Graph Convolution Network (GCN).** Due to the higher representation power of graph structure, GCN has demonstrated superior performance in several tasks, including image caption [8], text to image and human pose estimation [4]. In 3D computer vision, Wald *et al.* [40] proposed the first learning method that generated a semantic scene graph from a 3D point cloud. DGCNN [43] used a GCN-based network for point cloud feature extraction. Superglue [34] leveraged GCN to match two sets of local features by jointly searching correspondences and rejecting non-matchable points.

## 3. Approach

In this section, we will describe our depth-guided 6D pose regression network. We first introduce the relevant background. Then, we illustrate our network architecture which can learn the depth to refine the 6D pose.

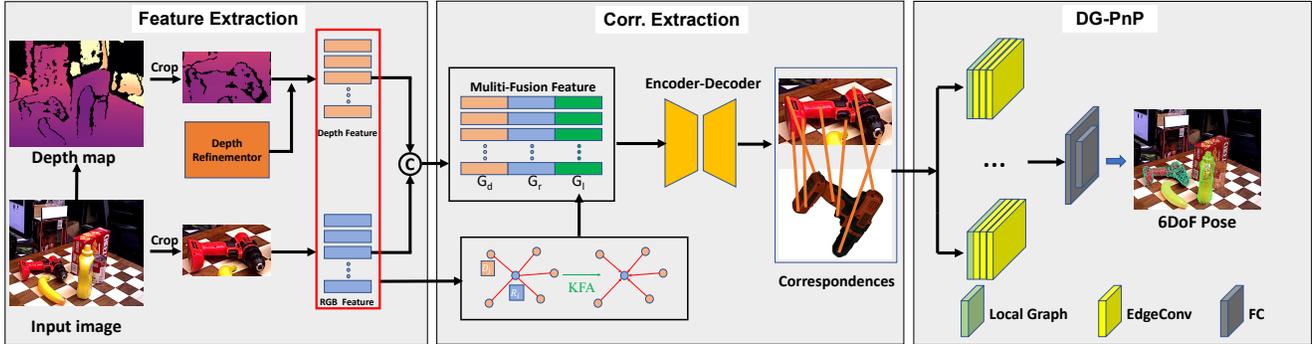


Figure 2. **Overview of our architecture.** Our framework consists of three building blocks: 1) a feature extraction network for depth and RGB features fusion, 2) a 2D-3D correspondences extraction network based on a deep voting-based network, and 3) a learnable PnP network named DG-PnP for 6D pose object estimation. KFA means K-NN Feature Aggregation,  $G_r$ ,  $G_d$  and  $G_l$  are RGB feature, depth feature and local feature, respectively.

### 3.1. Problem Formulation

Given an image, our task is to detect the objects and estimate the 6D pose of them. Here, we denote the image as  $I$ . Our goal is to estimate the rotation  $\mathbf{R} \in \mathbb{SO}(3)$  and translation  $\mathbf{t} = (t_x, t_y, t_z) \in \mathbb{R}^3$  that can transform the object from its object world coordinate system to the camera world coordinate system.

Fig 2 is the overview of our proposed method. We first learn depth information via an unsupervised depth estimation network. Afterwards, like GDR-Net [42] and PVNet [28], we locate each object in the image with the method of FCN [24]. According to the results of the segmentation, we crop the region of interest on depth map and RGB image, and fed them to a K-NN based feature aggregation (KFA) module to get the local features. Meanwhile, we use ResNet50 [13] to extract the 2D features of the image. Then, a dense fusion module is used to fuse the appearance features, geometry information and local features. Next, we take the fused feature as input of a 2D-3D correspondences prediction network to establish the 2D-3D correspondences. Finally, we directly regress the associated 6D object pose from the 2D-3D correspondences via our proposed differentiable DG-PnP.

Our framework builds upon keypoint-based methods. Given an image  $I$  and 3D models  $M = \{M_i | i = 1, \dots, N\}$ , our objective is to recover the unknown rigid transformation  $\{\mathbf{R}, \mathbf{t}\}$ . For the convenience of display, we assume that there is one target object in the image, we denote it as  $O$ . As shown in Fig 4, our goal is to predict the potential 2D location in  $I$  of the corresponding 3D keypoints of the model  $M$ .

### 3.2. Depth Estimation

Inspired by recent works [14, 41, 47, 48] based on RGB-D data and point cloud, we introduce depth information to make 2D-3D correspondences more robust and accurate.

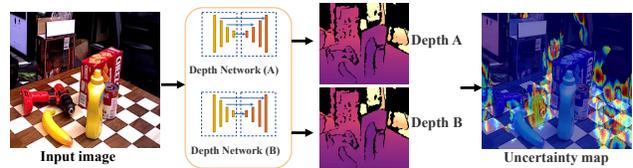


Figure 3. **Depth Uncertainty Measurement.**

However, these methods always need LIDAR or other sensors to get true depth information. Moreover, in a beforehand acquired RGB image, we usually can not obtain true depth information. Therefore, we use a network to predict the depth as an additional feature to supervise the 2D-3D correspondences estimation. With the development of monocular depth estimation, many depth estimation methods [11, 32, 44] have emerged. However, these methods are often used to estimate the depth information of large scenes, which is not good to directly estimate the depth map of 6D pose estimation scene. Therefore, in our work, we use uncertainty measurement to refine the estimated depth map.

### 3.3. Depth-Guided Edge Convolutional Network

The overview of our method is shown in Fig. 2. The keypoint localization is a voting-based architecture, which does not fully consider depth information. Therefore, we have made efforts in three directions to improve this strategy:

1. We leverage the uncertainty of estimated depth map on 6D object estimation scenes, we refine the depth map and reduce the influence of noise in the depth estimation process.
2. Before directly feeding RGB into CNN for establishing 2D-3D correspondences, we firstly predict the depth map and propose a K-NN Feature Aggregation (KFA) block to fuse cross-domain features.

3. We propose a learnable DG-PnP to replace the hand-crafted RANSAC/PnP in the two-stage 6d pose estimation pipeline.

**Depth refinement network (DRN).** Current monocular depth estimation methods are often applied to large outdoor scenes. Therefore, they are usually trained on large scenes dataset, such as KITTI. However, when we directly use these methods to estimate 6DoF scenes depth, in some areas, the fluctuations may be particularly large. The DRN aims to polish the quality of the depth map. As shown in Fig 3, it is composed of two different depth estimation networks, each network output a depth map  $D_A$  and  $D_B$ , respectively. We then calculate the difference between two depth maps, and define the area where the difference is over the threshold as an uncertain area. There are two ways to further handle these uncertain areas, one is directly remove them from the depth feature. The second way is to use their mean to replace the original depth. We choose the first way in this paper.

**Feature extraction.** This stage has two streams, one for depth estimation and the other for object segmentation. Depth estimation takes a color image as input and performs depth map prediction. Then, for each segmented object, we use the segmented object mask and the depth map to convert it to a 3D point cloud. To deal with multiple objects segmentation, previous works [17, 28, 41, 45] use existing detection or semantic segmentation algorithms. Similarly, we adopt FCN [24] to segment the input image. As for 3D feature extraction, some works [14, 41] convert the segmented depth pixels into a 3D point cloud, and the utilize 3D feature extractor [12, 29, 30] to extract geometric features. Although these methods are proved to be effective, they need to train additional 3D feature network. For more sufficient RGB-D fusion, we introduce KFA module. Consider a pixel in RGB image, denoted as  $p_i$ , and  $D_i = \{d_j | j = 1 \dots k\}$  is a depth set of the  $k$ -nearest neighbors of  $p_i$ , then we adopt a nonlinear function  $F_{p_i} = f(D_i, \theta_i)$  with a learnable parameter  $\theta_i$  to aggregate the local feature of  $p_i$ . As shown in Fig. 2, the resulting feature  $G = (G_r, G_d, G_l)$ .

**2D keypoint localization.** The 3D keypoints are selected from the 3D object model as in [14, 28]. Some methods [17, 31] choose the eight corners of the 3D bounding box. However, these points are virtual and 2D correspondences may locate outside the image. For the object closed to the boundary, this may lead to large errors, since the 2D correspondences are not in the image. Therefore, the keypoints should be selected on the object surface. We follow [28] and adopt the farthest point sampling (FPS) algorithm to select keypoints on object surface. At the end of this stage, we use a network based on [17] for 2D correspondences detection.

**Learning 6D pose from 2D-3D correspondences.** As shown in Fig. 4, given a set with  $n$  3D keypoints  $K =$

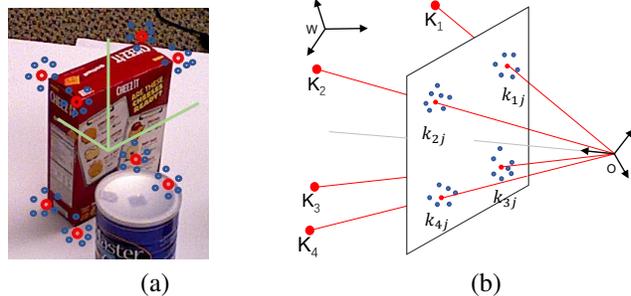


Figure 4. **2D-3D correspondences.** (a) Ground truth 2D correspondences (red ones) with their hypotheses (blue ones). (b) 2D correspondences projections on the camera plane. The camera and object coordinate systems are denoted by  $O$  and  $W$ , respectively.

$\{K_i | i = 1, \dots, n\}$  and each  $K_i$  corresponds to a set of 2D locations  $k = \{k_{ij} | j = 1, \dots, m\}$  in image. Our goal is to design a network to learn the rigid transformation  $(\mathbf{R}, \mathbf{t})$  from the established 2D-3D correspondences. DSAC [3] made RANSAC differentiable by soft argmax and probabilistic selection. Single-Stage [16] utilized a PointNet-like architecture to address this, however it can only handle sparse correspondences. GDR-Net [42] proposed a simple but effective patch-PnP module, where it depends on the dense correspondences maps that predicted by their network. To handle this, we propose a GCN based network to directly regress the 6D pose from the 2D-3D correspondences, which is described as follows

$$(\mathbf{R}, \mathbf{t}) = \mathcal{M}(K, k | \Theta), \quad (1)$$

where  $\mathcal{M}$  denotes the proposed DG-PnP with parameters  $\Theta$ .

Hu *et al.* [16] used an architecture similar to PointNet [29]. However, it only takes the 2D location as individual point and does not take into account the distribution property of 2D correspondences in the image. As mentioned above, we predict depth value of every pixel in the input image, therefore we can make full use of the geometric and location features of 2D correspondences. By revisiting the properties of 2D-3D correspondences, we find that the structure of the 2D correspondences is similar to a graph. As shown in Fig. 4, instead of taking individual points as input, we take the 2D correspondence cluster as a graph and feed it into our DG-PnP.

**Local Graph Construction.** As shown in Fig. 5,  $\mathcal{P} = \{p_i | i = 1 \dots m\}$  is a 2D correspondences cluster, we construct the local graph via  $k$ -nearest neighbor ( $k$ -NN) and denote it as  $\mathcal{G} = (\mathcal{P}, \mathcal{E})$ .  $\mathcal{P}$  and  $\mathcal{E} = p_i \leftrightarrow p_j$  are vertices and edges, respectively. Then, we compute edge features by aggregating all neighborhoods of  $p_i$  in  $\mathcal{P}$ .

**Edge-convolution.** Different from graph convolution network (GCN), our edge-convolution is a variant of CNN.

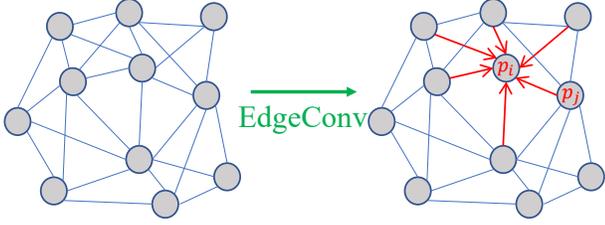


Figure 5. Local graph and edge convolution.

Considering a 2D correspondence cluster of  $m$  pixels with  $X$  dimension features, and denoting it as  $f = \{f_i | i = 1, \dots, m\}$ , we compute the local graph feature by our graph operation:

$$f'_i = \sum_{j=1}^m \lambda_j g_{\theta_i}(f_i, f_j), \quad (2)$$

where  $\lambda_j$  is a hyperparameter which is determined by the distance between  $k_i$  and  $k_j$ .  $g_{\theta}$  is a nonlinear function with a learnable parameters  $\theta$ . We adopt an asymmetric edge function proposed in [43]:

$$g_{\theta_i}(f_i, f_j) = \text{RELU}(\alpha_i \cdot (f_i - f_j) + \beta_i \cdot f_i), \quad (3)$$

where  $\theta_i = (\alpha_i, \beta_i)$  and  $\Theta = \{\theta_i | i = 1, \dots, m\}$  in Eq. 1. In this paper, we take the 3D coordinates and RGB information of  $k_i$  as features  $f_i$ , and the 3D coordinates can be transformed from depth using camera intrinsic. Therefore,  $X = 6$  in our network.

### 3.4. Loss Function and Pose Estimation

To train the proposed network, we introduce four loss functions  $\mathcal{L}_d$ ,  $\mathcal{L}_s$ ,  $\mathcal{L}_k$ , and  $\mathcal{L}_p$ . The total loss function is defined as

$$\mathcal{L} = \lambda_1 \mathcal{L}_d + \lambda_2 \mathcal{L}_s + \lambda_3 \mathcal{L}_k + \lambda_4 \mathcal{L}_p, \quad (4)$$

where  $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$  and  $\lambda_4$  are the weight coefficients.

$\mathcal{L}_d$  is the depth loss, and depth estimation module is built upon MonoDepth2 [11]:

$$\mathcal{L}_d = \mu L_{dp} + \lambda L_{ds}, \quad (5)$$

where  $L_{dp}$  is photometric loss, and  $L_{ds}$  is edge-aware smoothness. Due to space limitations, further details can refer to [11].

$\mathcal{L}_s$  is the segmentation loss, which is used to constrain the segmentation task and extract the target object from the image. Here we choose the Focal Loss according to [23].

$\mathcal{L}_k$  is the keypoint matching loss, which is used to constrain the 2D-3D correspondences. As shown in Fig. 4, we seek to predict 2D keypoints location in the image and we define the loss function as:

$$\mathcal{L}_k = \frac{1}{M} \sum_{i=1}^n \sum_{j=1}^m \|kp_{ij} - kp_i^*\|, \quad (6)$$

where  $kp_i^*$  is the ground truth 2D keypoint location,  $n$  is the number of 3D keypoints,  $m$  is the number of 2D correspondences of  $kp_i$ ,  $M = m \times n$  is the number of total 2D correspondences predicted by our network in the image.

$\mathcal{L}_p$  is the final pose estimation loss, which is used to constrain the final 6DoF pose parameters. Inspired by PoseCNN [45] and DeepIM [21], we design  $\mathcal{L}_p$  as

$$\mathcal{L}_p = \frac{1}{n} \sum_{i=1}^n \|(\mathbf{R}^* \mathbf{p}_i + \mathbf{t}^*) - (\mathbf{R} \mathbf{p}_i + \mathbf{t})\|. \quad (7)$$

where  $\mathbf{R}^*$  and  $\mathbf{t}^*$  are the estimated rotation matrix and translation vector,  $\mathbf{R}$  and  $\mathbf{t}$  are the ground-truth ones.

Our network is a multi-task network including calculations of output depth map, segmentation mask, 3D-2D correspondences, and 6DoF pose parameters like the current SOTA methods. More generally, when there are multiple target objects in the image, we can estimate the poses of these target objects simultaneously, and the results are given in the experimental section.

## 4. Experiments

In this section, we conduct experiments to prove the effectiveness of DGECON. We evaluate our DGECON on several common benchmark datasets. For direct comparison to classic PnP and some learning PnP, we set up several experiments following [16, 42] on a synthetic sphere dataset to verify the proposed DG-PnP. Further, we conduct an ablation study to discuss the effectiveness of each component in the proposed method.

### 4.1. Datasets

#### 4.1.1 Synthetic Sphere Dataset.

As in Single-Stage [16], we create the exact synthetic 3D-to-2D correspondences using a virtual calibrated camera, with image size of  $640 \times 480$ , focal length of 800, and principal point at the image center. However, Single-Stage does not require color information, so their background is pure. As discussed in Sec. 3, our network will fully extract local features, including location and color. So we add a gradient background to their synthetic dataset, and the other parameter settings are the same with Single-Stage, as shown in Fig. 7.

#### 4.1.2 YCB-V Dataset.

This dataset is proposed by [5, 45] and consists of 21 YCB objects with different shapes and textures. 92 RGB-D videos of the subset of objects were captured and annotated with 6D pose and instance semantic mask. The varying lighting conditions, significant image noise, and occlusions make this dataset challenging. As in PoseCNN [45],

2D-3D extractor	PnP type	Ape	Can	Cat	Driller	Duck	Eggbox <sup>s</sup>	Glue <sup>s</sup>	Holepun	Mean
DGECON(Ours)	DG-PnP(Ours)	54.3	75.9	22.4	77.5	51.2	57.8	66.9	63.2	58.7
	PointNet-like PnP [16]	44.4	71.3	18.5	71.6	48.6	51.3	59.1	60.3	53.1
	Patch-PnP [42]	51.2	74.6	21.6	73.4	48.5	56.9	65.1	61.4	56.6
	RANSAC-based PnP [20]	41.3	66.5	14.3	65.4	44.1	48.9	55.4	56.2	49.0
	BPnP [6]	46.2	73.3	19.5	72.4	46.2	52.1	61.4	56.2	53.4
PVNet [28]	DG-PnP(Ours)	23.4	68.9	23.2	72.2	27.8	55.1	53.2	47.2	46.4
	PointNet-like	19.2	65.1	18.9	69.0	25.3	52.0	51.4	45.6	43.3
	Patch-PnP	14.4	55.3	14.9	68.2	22.1	45.9	49.4	41.3	38.9
	RANSAC-based PnP	15.8	63.3	16.7	65.7	25.2	50.3	49.6	36.1	40.8
	BPnP	21.4	45.3	12.7	64.3	21.4	42.1	44.5	38.7	36.3
SegDriven [17]	DG-PnP(Ours)	17.5	51.4	15.9	57.9	20.6	31.8	43.2	39.6	34.7
	PointNet-like	14.8	45.5	12.1	54.6	18.3	30.2	45.8	37.4	32.3
	Patch-PnP	9.8	36.9	14.6	57.3	11.6	28.3	42.3	32.4	28.4
	RANSAC-based PnP	12.1	39.9	8.2	45.2	17.2	22.1	35.8	36.0	27.0
	BPnP	15.6	47.8	14.5	51.3	14.8	30.5	26.4	32.1	29.1
GDR-Net [42]	DG-PnP(Ours)	37.5	78.5	26.8	70.6	42.9	56.8	50.4	56.4	52.5
	PointNet-like PnP	17.9	65.3	18.6	62.8	31.5	48.6	36.7	49.2	41.3
	Patch-PnP	39.3	79.2	23.5	71.3	44.4	58.2	49.3	58.7	53.0
	RANSAC-based PnP	20.9	67.5	23.9	66.1	34.9	53.4	42.3	54.3	45.4
	BPnP	35.5	74.2	21.5	67.4	36.9	51.4	45.8	51.1	48.0

Table 1. **Ablation Study.** Results for different versions of our model with comparison to some baseline models. We evaluate the impact of the DGECON, and DG-PnP. (s) denotes symmetric objects, metrics indicated by light red is the best result. We report the Average Recall (%) of ADD(-S) on LM-O dataset.

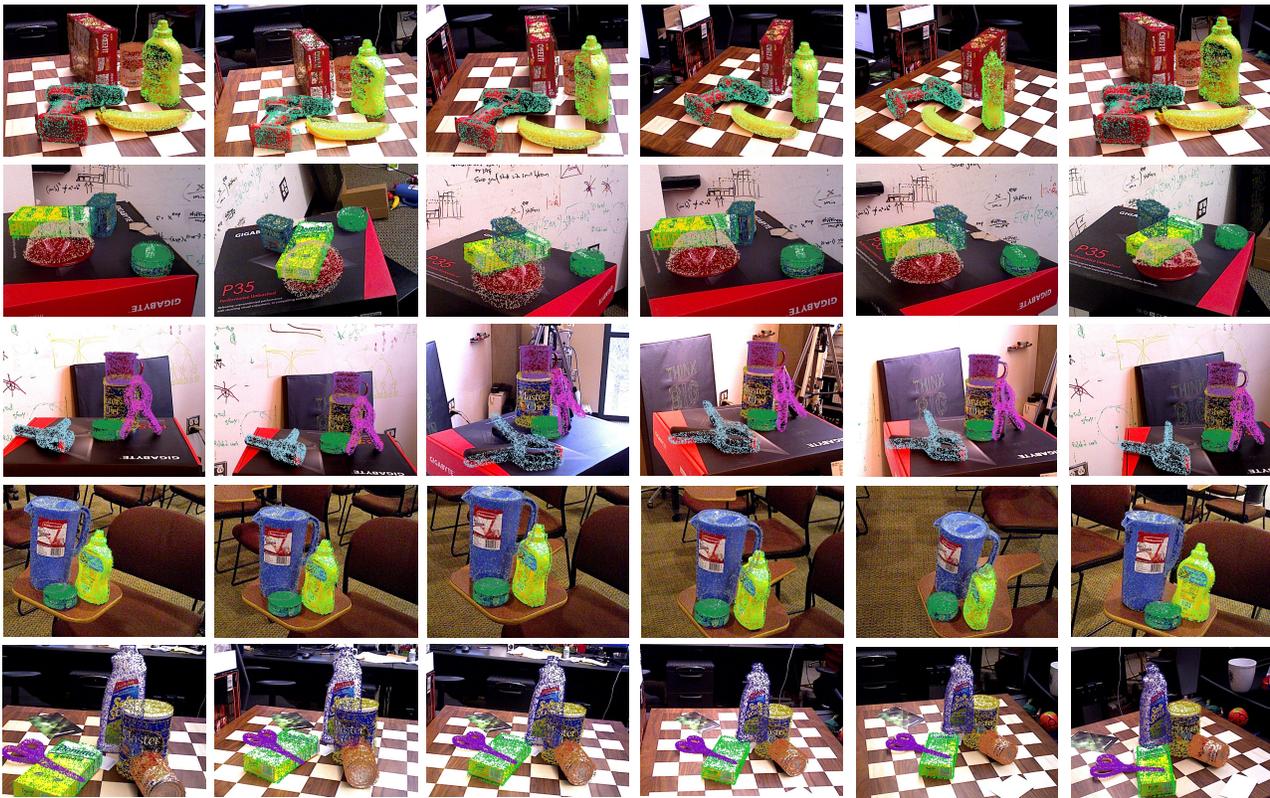


Figure 6. **Qualitative results on YCB-V dataset.** Here we show visualizations of results on YCB-V dataset. Points on different meshes in the same scene are in different colors which projected back to the image after being transformed by the predicted pose.

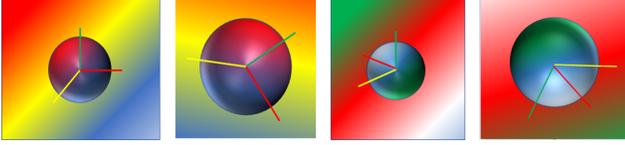


Figure 7. **Synthetic data.** We create synthetic data as in [16], but we add background on theirs.

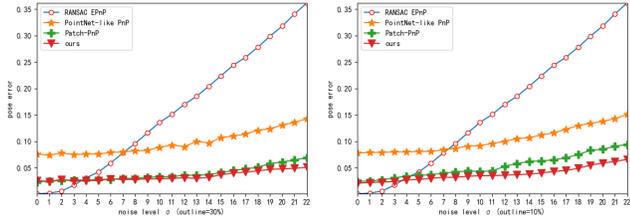


Figure 8. **Comparison with PnP variants .** We compare our method with EPnP [20], PointNet-like PnP [16] and Patch-PnP [42]. Our method performs better than PointNet-like PnP all time, and as the noise increases our method is much more accurate and robust than EPnP. The pose error is calculated by ADD.

we split the dataset into 80 videos for training and a set of 2,949 keyframes chosen from the rest 12 videos for testing.

#### 4.1.3 LM-O Dataset.

This dataset [2] is a standard benchmark for object 6D pose estimation and contains 13 low-textured objects in 13 videos, annotated 6D pose and instance mask. The main challenges of LM-O are the chaotic scenes, texture-less objects, and lighting variations. In this work, we follow prior works to handle this dataset, and we also add synthesised images into our training set as in [45].

### 4.2. Evaluation metrics

For comparison, we evaluate our method with two common metrics: the average distance (ADD) [45] and the 2D reprojection error (REP) [17].

**ADD** uses the average distance between the 3D model points transformed using the predicted pose and those obtained with the ground-truth one. When the distance is less than 10% of the model’s diameter, it claims that the estimated pose is correct. We follow [16, 42] and evaluate the symmetric object by ADD(-S) metric, which measures the deviation to the closet model point. Denote the predicted pose as  $[\mathbf{R}^*, \mathbf{t}^*]$  and the ground truth pose as  $[\mathbf{R}, \mathbf{t}]$ :

$$\text{ADD} = \frac{1}{m} \sum_{x \in \mathcal{O}} \|(Rx + t) - (R^*x + t^*)\| \quad (8)$$

$$\text{ADD-S} = \frac{1}{m} \sum_{x_1 \in \mathcal{O}} \min_{x_2 \in \mathcal{O}} \|(Rx_1 + t) - (R^*x_2 + t^*)\| \quad (9)$$

where  $x$  is a vertex of totally  $m$  vertices on object mesh  $\mathcal{O}$ . When evaluating on YCB-V, we also compute the AUC (area under curve) of ADD(-S) by varying the distance threshold with a maximum of 10 cm [45].

**REP** computes the mean distance between the projections of 3D model points given the estimated and the ground truth pose. When the REP is below 5 pixels, we claim that the estimated pose is correct.

For each metric, we use the symmetric version for symmetric objects, which we denote by a superscript (s).

### 4.3. Comparison with State-of-the-arts

We compare with the state-of-the-art works on YCB-V and LM-O datasets. It is worth mentioning that we also make a comparison with the RGB-D based methods to verify the effectiveness of our depth estimation network.

#### 4.3.1 Performance on LM-O dataset.

Tab. 2 shows the results of DGE CN compared with the state-of-the-art monocular methods on Occlusion LM-O dataset. Our DGE CN is comparable to [7, 21, 42] and outperforms [16, 28]. Tab. 5 presents the results of compared with RGB-D based methods. Moreover, in some scenes, the proposed method even outperforms the RGB-D based methods.

#### 4.3.2 Performance on YCB-V

Tab. 3 shows the evaluation results for YCB-V dataset. It shows that our model is comparable to the state-of-the-arts [19, 42] and even outperforms the refinement-based method [21]. Fig. 6 demonstrates qualitative results on YCB-V.

### 4.4. Ablation study

In this section, we would like to discuss the following questions: (1) How does the DG-PnP compare to the hand-crafted PnP and other learnable PnP? (2) Does the learned depth improve the final pose estimation? (3) Is the DGE CN effective with PnP variants?

**Comparison to PnP variants.** We take 20K synthetic images for training and 2K images for testing. While training, we randomly add 2D noise with variance  $\sigma$  in the range of  $[0, 15]$  and create outliers with 10% and 30%. Comparison in synthetic is critical, because it can directly compare our DG-PnP with PnP variants and ignore the influence of the keypoints detection methods. Fig. 8 shows the results at different noise levels, compared with EPnP [20], PointNet-like PnP [16] and Patch-PnP [42]. While handcrafted PnP is more accurate when the noise is minimal, learnable PnP methods are more robust to noise, and they are more accurate when the noise increasing. Moreover, DG-PnP is

Method	PoseCNN	PVNet	Single-Stage	HybridPose	GDR-Net	SO-Pose	DeepIM <sup>R</sup>	DPOD <sup>R</sup>	Ours
Ape	9.6	15.8	19.2	20.9	41.3	46.3	<b>59.2</b>	-	50.3
Can	45.2	63.3	65.1	75.3	71.1	<b>81.1</b>	63.5	-	75.9
Cat	0.9	16.7	18.9	24.9	23.5	18.2	26.2	-	<b>26.4</b>
Driller	41.4	65.7	69.0	70.2	54.6	71.3	55.6	-	<b>77.5</b>
Duck	19.6	25.2	25.3	27.9	41.7	43.9	52.4	-	<b>54.2</b>
Eggbox <sup>s</sup>	22.0	50.2	52.0	52.4	40.2	46.6	<b>63.0</b>	-	57.8
Glue <sup>s</sup>	38.5	49.6	51.4	53.8	59.5	63.3	<b>71.7</b>	-	66.9
Holepun	22.1	36.1	45.6	54.2	52.6	62.9	52.5	-	<b>60.2</b>
Mean	24.9	40.8	43.3	47.5	47.4	54.3	55.5	47.3	<b>58.7</b>

Table 2. Quantitative comparison on known categories of LM-O dataset with state-of-the-art RGB methods with the metric as ADD(-S), (R) stands for Refinement. All methods trained with *real* + *syn* data.

Method	Ref.	ADD(-S)	REP-5px	AUC of ADD-S
PoseCNN [45]	✗	21.3	3.7	75.9
GDR-Net [42]	✗	60.1	-	<b>91.6</b>
SO-Pose [7]	✗	56.8	-	90.9
PVNet [28]	✗	-	47.4	73.4
SegDriven [17]	✗	39.0	30.8	-
Single-Stage [16]	✗	53.9	48.7	-
DeepIM [21]	✓	-	-	88.1
CosyPose [19]	✓	-	-	89.8
Ours	✗	<b>60.6</b>	<b>50.3</b>	90.9

Table 3. Evaluation with state-of-the-art RGB methods on YCB-V. Ref. stands for Refinement.

Corr. Extractor	DG-PnP	ADD	AUC of ADD-S
✓	✓	58.7	90.9
✓	✗	53.2	83.5
✗	✓	50.6	81.3
✗	✗	41.3	75.3

Table 4. **Ablation on Depth Map.** ✓ denotes test with depth map and ✗ denotes test without depth map.

Method	ADD(-S)	REP-5px	AUC of ADD-S
Implicit ICP [37]	64.7	-	-
SSD-6D ICP [18]	79.0	-	91.6
PointFusion [46]	-	73.7	73.4
DenseFusion [41]	86.2	30.8	-
PVN3D [14]	53.9	99.4	-
Ours	60.6	50.3	90.9

Table 5. Evaluation with state-of-the-art RGB-D methods on YCB-V.

significantly more robust and accurate than PointNet-like PnP, and comparable with Patch-PnP. Because DG-PnP and Patch-PnP both take into account the geometric and topol-

ogy features.

**Ablation on depth map.** As mentioned above, depth information plays a significant role in 6D pose regression. Furthermore, we train our DGECN by discarding the depth estimation. The depth information is used in both correspondence extraction and DG-PnP, so we setup a ablation study on it. As shown in Tab. 4. DGECN is significantly more robust with depth prediction.

**Effectiveness of each component.** As shown in tab. 1, we demonstrate the effectiveness of each component of the proposed method by combining our components with different state-of-the-art methods. For DGECN, we replace the DG-PnP in our architecture with PnP variants [6, 16, 42]. DGECN demonstrates a competitive performance with different PnP methods. Moreover, it is even better than Single-Pose combined with the PointNet-like PnP. As for DG-PnP, we replace the PnP variants in some two-stage methods with DG-PnP.

## 5. Conclusion

In this work, we propose a novel depth-guided network for monocular 6D object pose estimation. The core idea is to utilize geometric and topology information, and jointly handles 2D keypoint detection and 6D pose estimation. Then, we delve into 2D-3D correspondences and observe that graph structure can better model the feature of keypoint distributions. Furthermore, we propose a dynamic graph PnP for learning 6D pose to replace the handcrafted PnP. Thus, our approach is a real-time, accurate and robust monocular 6D object pose estimation method.

## 6. Acknowledgments

This work is partially supported by the Key Technological Innovation Projects of Hubei Province (2018AAA062), National Nature Science Foundation of China (NSFC No.61972298) and Wuhan University Huawei GeoInformatics Innovation Lab.

## References

- [1] Alex M Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001. **1**
- [2] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014. **2, 7**
- [3] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac - differentiable ransac for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. **2, 4**
- [4] Yujun Cai, Liuhaohao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2272–2281, 2019. **2**
- [5] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015. **2, 5**
- [6] Bo Chen, Alvaro Parra, Jiwei Cao, Nan Li, and Tat-Jun Chin. End-to-end learnable geometric vision by backpropagating pnp optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8100–8109, 2020. **6, 8**
- [7] Yan Di, Fabian Manhardt, Gu Wang, Xiangyang Ji, Nassir Navab, and Federico Tombari. So-pose: Exploiting self-occlusion for direct 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12396–12405, October 2021. **2, 7, 8**
- [8] Xinzhi Dong, Chengjiang Long, Wenju Xu, and Chunxia Xiao. Dual graph convolutional networks with transformer and curriculum learning for image captioning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2615–2624, 2021. **2**
- [9] Yanping Fu, Qingan Yan, Jie Liao, and Chunxia Xiao. Joint texture and geometry optimization for rgb-d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5950–5959, 2020. **1**
- [10] Yanping Fu, Qingan Yan, Jie Liao, Huajian Zhou, Jin Tang, and Chunxia Xiao. Seamless texture optimization for rgb-d reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2021. **1**
- [11] Clement Godard, Oisín Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, pages 3827–3837, 2019. **3, 5**
- [12] Meng-Hao Guo, Junxiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: point cloud transformer. *CoRR*, abs/2012.09688, 2020. **4**
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. **3**
- [14] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11632–11641, 2020. **1, 3, 4, 8**
- [15] Tomas Hodan, Daniel Barath, and Jiri Matas. Epos: estimating 6d pose of objects with symmetries. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11703–11712, 2020. **1, 2**
- [16] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2930–2939, 2020. **2, 4, 5, 6, 7, 8**
- [17] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3385–3394, 2019. **1, 2, 4, 6, 7, 8**
- [18] 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 *Proceedings of the IEEE international conference on computer vision*, pages 1521–1529, 2017. **8**
- [19] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020. **7, 8**
- [20] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Eppnp: An accurate o(n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009. **6, 7**
- [21] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018. **5, 7, 8**
- [22] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7678–7687, 2019. **1**
- [23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. **5**
- [24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. **3, 4**
- [25] Fabian Manhardt, Diego Martin Arroyo, Christian Rupprecht, Benjamin Busam, Tolga Birdal, Nassir Navab, and Federico Tombari. Explaining the ambiguity of object detection and 6d pose from visual data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6841–6850, 2019. **1**
- [26] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018. **2**

- [27] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7668–7677, 2019. [1](#), [2](#)
- [28] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019. [1](#), [2](#), [3](#), [4](#), [6](#), [7](#), [8](#)
- [29] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. [2](#), [4](#)
- [30] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. [4](#)
- [31] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3836, 2017. [1](#), [4](#)
- [32] Michael Ramamonjisoa, Michael Firman, Jamie Watson, Vincent Lepetit, and Daniyar Turmukhambetov. Single image depth prediction with wavelet decomposition. pages 11089–11098, June 2021. [3](#)
- [33] Denys Rozumnyi, Jan Kotera, Filip Sroubek, and Jiri Matas. Sub-frame appearance and 6d pose estimation of fast moving objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6778–6786, 2020. [1](#)
- [34] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. [2](#)
- [35] Jianzhun Shao, Yuhang Jiang, Gu Wang, Zhigang Li, and Xiangyang Ji. Pfrl: Pose-free reinforcement learning for 6d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11454–11463, 2020. [1](#)
- [36] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 431–440, 2020. [1](#)
- [37] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the european conference on computer vision (ECCV)*, pages 699–715, 2018. [8](#)
- [38] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. [2](#)
- [39] Kentaro Wada, Edgar Sucar, Stephen James, Daniel Lenton, and Andrew J Davison. Morefusion: multi-object reasoning for 6d pose estimation from volumetric fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14540–14549, 2020. [1](#)
- [40] Johanna Wald, Helisa Dharmo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. [2](#)
- [41] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3343–3352, 2019. [1](#), [3](#), [4](#), [8](#)
- [42] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16611–16621, June 2021. [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [43] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. [2](#), [5](#)
- [44] Jamie Watson, Michael Firman, Gabriel Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *ICCV*, pages 2162–2171, 2019. [3](#)
- [45] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. [1](#), [2](#), [4](#), [5](#), [7](#), [8](#)
- [46] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 244–253, 2018. [8](#)
- [47] Wenxiao Zhang and Chunxia Xiao. Pcan: 3d attention map learning using contextual information for point cloud based retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12436–12445, 2019. [3](#)
- [48] Wenxiao Zhang, Qingan Yan, and Chunxia Xiao. Detail preserved point cloud completion via separated feature aggregation. In *European Conference on Computer Vision*, pages 512–528. Springer, 2020. [3](#)
- [49] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. [2](#)