Category-Level 6D Object Pose Estimation in the Wild: A Semi-Supervised Learning Approach and A New Dataset

Yang Fu **Xiaolong Wang**
UC San Diego

Abstract

6D object pose estimation is one of the fundamental problems in computer vision and robotics research. While a lot of recent efforts have been made on generalizing pose estimation to novel object instances within the same category, namely category-level 6D pose estimation, it is still restricted in constrained environments given the limited number of annotated data. In this paper, we collect Wild6D, a new unlabeled RGBD object video dataset with diverse instances and backgrounds. We utilize this data to generalize category-level 6D object pose estimation in the wild with semi-supervised learning. We propose a new model, called **Re**ndering for **Po**se estimation network (**RePoNet**), that is jointly trained using the free ground-truths with the synthetic data, and a silhouette matching objective function on the real-world data. Without using any 3D annotations on real data, our method outperforms state-of-the-art methods on the previous dataset and our Wild6D test set (with manual annotations for evaluation) by a large margin. Project page with Wild6D data: `https://oasisyang.github.io/semi-pose/`.

1 Introduction

Estimating the 6D object pose is one of the core problems in computer vision and robotics. It predicts the full configurations of rotation, translation and size of a given object, which has wide applications including Virtual Reality (VR) [2], scene understanding [30], and [42, 57, 31, 49]. There are two directions in 6D object pose estimation. One is performing instance-level 6D pose estimation, where a model is trained to estimate the pose of one exact instance with an existing 3D model [13, 34, 22, 50, 32, 5, 14]. However, learning instance-level model restricts its generalization ability to unseen objects. To achieve generalization to unseen instance, another direction is recently proposed to perform category-level 6D pose estimation using one model [48, 3, 41, 29, 6]. However, the large appearance and shape variance across instances largely increase the difficulty in learning.

To overcome this limitation, Wang *et al*. [48] take the initial step to collect real-world dataset and annotations for category-level 6D pose estimation. Combining synthetic data with free ground-truth annotations, they show the learned model can be generalized to unseen objects within the same category. While this result is encouraging, the generalization ability of the model is still limited by the number and the diversity of the data due to the challenges in annotating 6D object poses. Specifically, only 8,000 images across 13 scenes are collected and annotated from the dataset proposed in [48]. Thus it is still very challenging to generalize 6D pose estimation on diverse objects in complex scenes.

In this paper, we propose to generalize category-level 6D object pose estimation in the wild. To achieve this goal, we introduce a new dataset and a new semi-supervised learning approach. Our key insight is that, while annotating the 6D object pose is challenging, collecting the RGBD videos for these objects without labels is much easier and more affordable. On the other hand, there are infinite ground-truth annotations for synthetic data which come for free. We propose to leverage the benefits from both sides. We first collect a rich object-centric video dataset with diverse backgrounds and

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

https://doi.org/10.52202/068431-1992

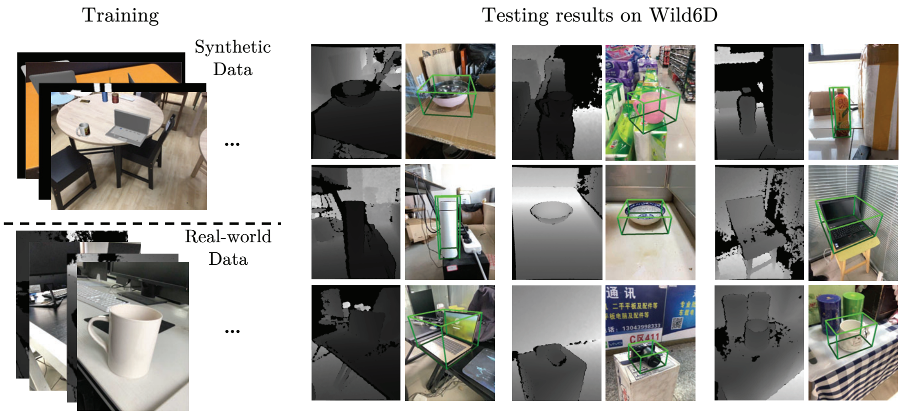

Figure 1: **Left**: we train our model with both synthetic data and real-world data under our proposed semi-supervised setting. **Right**: during inference, given the RGBD images, the object pose can be estimated precisely. Green bounding boxes show the 3D bounding boxes projection results on 2D images.

object instances using an RGBD camera. We train our model jointly using synthetic data with the free ground-truth 6D pose annotations and the unlabeled real-world RGBD videos via a silhouette matching objective. In this way, our pose estimation model can be generalized to in-the-wild data with minimum human labor. We collect an RGBD video dataset for 6D object pose estimation in the wild, namely *Wild6D*. Each video in the dataset shows multiple views of one or multiple objects (see examples in Fig. 1). In total, there are 5166 videos (> 1.1 million images) over 1722 object instances and 5 categories, which is significantly (300x) larger than the previous 6D object pose estimation dataset [48]. For evaluation, we annotate 486 videos over 162 objects.

Given this dataset, we design a novel model for semi-supervised learning, called **Re**ndering for **Po**se estimation **Net**work (RePoNet). The RePoNet is composed of two branches of networks with a *Pose Network* to estimate the 6D object pose and a *Shape Network* to estimate the 3D object shape. Given an RGBD image of an object, our Pose Network first estimates the Normalized Object Coordinate Space (NOCS) map [48], which will be integrated with lower-layer features to regress the object 9D pose (rotation, translation, and size parameters). Meanwhile, the Shape Network takes a category-level 3D shape prior as an input, and estimates the object shape for the current input instance. During training, we utilize both the synthetic data with the full ground-truths and the real-world data from Wild6D with foreground segmentation masks (obtained by Mask R-CNN [12]). Given the inputs with synthetic data, we apply the regression losses on both the NOCS map, 6D pose parameters, and object shape. Additionally, given the estimated pose and object shape, we perform differentiable rendering to obtain an object mask projected in 2D. A silhouette matching objective is proposed to compare the difference between the projected mask and the ground-truth mask. With the real RGBD data, although we do not have the 3D ground-truths, we can still learn with the silhouette matching objective by comparing the projected mask against the foreground segmentation. In this way, the gradients are back-propagated through the 6D pose and object shape to adjust both Pose Network and Shape Network for real RGBD data. During inference, we first apply our Pose Network to estimate the NOCS map with a given image. With the NOCS map output, the object pose can be computed by solving the Umeyama algorithm [43]. Some estimation results on Wild6D can be found in Fig. 1.

In our experiments, we first evaluate our approach with the dataset proposed in [48]. Our RePoNet shows a improvement over state-of-the-art approaches when trained in a fully-supervised setting (using 3D ground-truths from both real and synthetic data). By using the real-world data without its 3D ground-truth with semi-supervised learning, our performance can still be on par with previous approaches using full annotations. We then experiment with our Wild6D dataset and consistently show a large performance gain over the baselines on in-the-wild objects. We highlight our main contributions as follows:

- A new large-scale dataset Wild6D of object-centric RGBD videos in-the-wild for category-level 6D object pose estimation.
- A semi-supervised approach with RePoNet which leverages both synthetic data and real-world data without 3D ground-truths for category-level 6D object pose estimation.

https://doi.org/10.52202/068431-1992

- Our approach outperforms baselines by a large margin, especially when deployed on in-the-wild objects. We commit to **release our dataset, annotation and code** for benchmarking future research.

2 Related Work

Category-level 6D Object Pose Estimation. Recently, researchers have proposed to learn a single model for one category of objects instead of just one instance on pose estimation. For example, Wang *et al*. [48] propose a canonical shape representation of different objects called Normalized Object Coordinate Space (NOCS) to handle the instance variations. The object pose and size are calculated by the Umeyama algorithm [43] with predicted NOCS map and the observed points. Follow-up work using the NOCS representation has focused on improving the shape priors [41] and incorporating direct regression of object pose and size [3, 25, 6]. While these approaches can be deployed on unseen object instances, the generalization ability of these models is still limited by the scale and the diversity of real-world annotated data in the REAL275 dataset [48]. Only 13 scenes with 18 real objects in total are presented in REAL275. In contrast, we propose a semi-supervised learning approach with RePoNet, which leverages a new large-scale unlabeled dataset Wild6D for training. Both our method and the new dataset are the key components that lead to our goal of generalizing category-level 6D pose estimation in the wild.

Large-scale 3D Object Datasets. Various large-scale 3D datasets have been proposed for different tasks including reconstruction and pose estimation. For example, the Objectron dataset are collected and studied with large-scale object-centric videos and 3D annotations [1, 15, 26]. However, there are no depth images provided in Objectron, which might lead to ambiguities in 6D object pose estimation. Similarly, the recent proposed CO3D [37] dataset with diverse instances in different categories is also collected without recording the depth. While we can perform 3D reconstruction or generate the depth map using COLMAP [38], the error in predicted depth makes it difficult to be used for 6D pose estimation. Different from these datasets, the 3DScan [9] dataset records RGBD videos of different categories of objects. However, most objects are heavily occluded by hands or only partially visible with large objects like cars. The diversity of objects and scenes is also relatively low given limited human labor. Thus 3DScan is not suitable for pose estimation tasks. In this paper, we propose *Wild6D*, a dataset with RGBD videos containing diverse objects taken with diverse backgrounds. While the training set is not labeled, we provide 6D pose annotations for the test videos. To the best of our knowledge, Wild6D is the largest RGBD dataset for 6D object pose estimation in the wild.

Rendering in Object Pose Estimation. One effective refinement method for 6D object pose estimation is via rendering [24, 55, 20, 16]. For instance, Iwase *et al*. [16] propose to utilize differentiable rendering and learn the texture of a 3D model. However, it still depends on a pre-defined CAD model with a high-quality texture map for instance-level pose estimation and it is not feasible for category-level. Additionally, object pose estimation can be conducted via Analysis-by-Synthesis [44, 8, 52, 46, 40], but they usually require gradient descent for optimization leading to relatively slower inference speed. Critically, these works only focus on the instance-level setting and none of them can generalize to different instances. Inspired by recent work on 3D reconstruction [17, 23, 7, 18], our RePoNet learns the object shape and pose (including both NOCS map and R/T/S parameters) simultaneously using differentiable rendering to provide a loss function during learning [28], which allows the gradients to backprop through the whole network for training. While Manhardt *et al*. [29] also utilize differentiable rendering to improve category-level 6D pose, it is only applied in test time to adjust the pose, instead of training the network end-to-end. We show substantial improvement over all baselines training with RePoNet.

3 Wild6D Dataset

Existing Category-level 6D Object Pose Datasets. NOCS [48] is the most common dataset for category-level 6D object pose estimation, which consists of the synthetic CAMERA25 dataset and the real-world REAL275 dataset. The synthetic CAMERA25 dataset contains 300,000 RGBD images of 1,085 object instances from 6 categories for training and evaluation. The REAL275 dataset shares the same categories with CAMERA25 but is more challenging under the real-world background. It contains 4,300 RGBD images of 7 scenes for training and 2,750 images of 6 scenes for testing. Due to the limited scale and diversity of real data, the learned models from this dataset cannot generalize to in-the-wild scenarios. **Objectron** [1] is a large-scale dataset for object pose estimation and tracking.

https://doi.org/10.52202/068431-1992

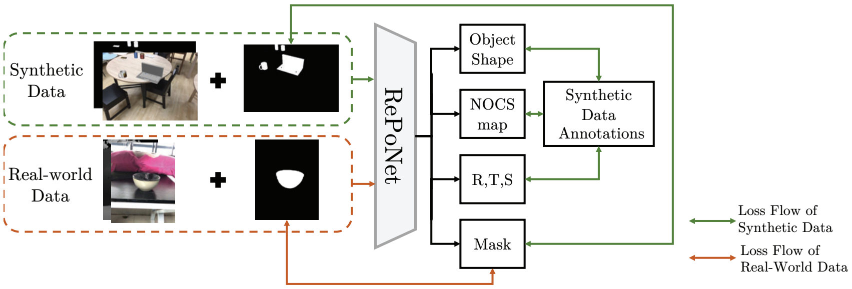
Synthetic Data
Real-world Data
RePoNet
Object Shape
NOCS map
R,T,S
Mask
Synthetic Data Annotations
Loss Flow of Synthetic Data
Loss Flow of Real-World Data

Figure 2: Our semi-supervised approach. For the synthetic data, we supervise it with all the annotations. While for the real-world data, we train it by comparing the binary mask generated by the rendering module with the object foreground segmentation.

Datasets	RGBD	Real	#Categories	#instances	#images
Objectron [1]		✓	9	18K	4M
CAMERA25 [48]	✓		6	184	300K
REAL275 [48]	✓	✓	6	24	8k
Wild6D	✓	✓	5	1.8K	1M

Table 1: **Comparison between Wild6D and existing datasets.** Wild6D significantly scale up the number of images, object instances, complexity of the scenes compared to previous RGBD datasets.

Different from NOCS [48], it consists of a collection of short object-centric video clips captured in the real world. However, no depth maps are provided in Objectron, which might lead to ambiguities in pose estimation.

Wild6D Collection. To achieve a categorical 6D pose for real objects, we collect a new large-scale RGBD dataset, named *Wild6D*. Each video in the Wild6D is recorded via the iPhone front camera showing multiple views of objects where RGB images and the corresponding depth images and point cloud are captured simultaneously. The videos are captured by different turkers with their own iPhones to guarantee the diversity of instances and background scenes. Three videos are taken for each object under different scenes. In total, Wild6D consists of 5,166 videos (>1.1 million images) over 1722 different object instances and 5 categories , *i.e.*, *bottle*, *bowl*, *camera*, *laptop*, and *mug*. Among this data, we split 486 videos of 162 instances to use them as the test set. Table 1 summarizes the statistics of Wild6D comparing previous datasets: Wild6D significantly improves the number of images, object instances, and scene complexity.

Wild6D Annotation. To annotate more than 10,000 images in the testing set efficiently, we propose a tracking-based annotation pipeline. Inside a video, we manually annotate the 6D object poses every 50 frames as keyframes. Given the annotation of the keyframe, we implement TEASER++ [51] together with colored ICP [33] to achieve the registration between the keyframe and the following frame and compute the transformation matrix. The ground-truth object pose of the following frame can be obtained by applying the transformation matrix to the keyframe annotation. Following this pipeline, we can obtain accurate ground-truths by only labeling around 5 keyframes per video.

4 Proposed Method

We propose the **Re**ndering for **Po**se estimation network (RePoNet) using both synthetic data and large-scale unlabeled real-world data in a semi-supervised manner.

RePoNet overview. The RePoNet is composed of two networks including the *Pose Network* to directly estimate the object 6D pose parameters (with the NOCS map as an intermediate representation) and the *Shape Network* to reconstruct the object shape. The outputs from both networks can go through a differentiable rendering [28] module which outputs a segmentation mask.

Semi-supervised learning setting. As shown in Fig. 2, we utilize two sets of data for semi-supervised learning: (i) synthetic data with full annotations including NOCS maps, CAD models, foreground segmentation, and 6D pose parameters, denoted as $\mathbf{D}_{\text{syn}}$; and (ii) a large-scale real-world RGBD dataset Wild6D with estimated foreground masks (using pre-trained Mask R-CNN [12]), denoted as $\mathbf{D}_{\text{real}}$. Both the synthetic data and the real-world data are utilized to train our RePoNet jointly. For the synthetic data, all the ground-truths are used as supervision signals for the *Pose Network* and the *Shape Network*. While for the real-world data, we directly feed the outputs from the *Pose Network* and the *Shape Network* into the differentiable rendering module to generate the binary mask, then train the whole RePoNet by comparing the rendered mask with the object foreground mask (silhouette matching loss). While there can be a more delicate way to deal with the domain gap between simulation and real, we find the direct sharing of parameters during training already provides a simple yet effective solution for generalization.

One important contribution of this framework is to make the whole procedure with RePoNet differentiable, using the Shape Network, and a ConvNet to connect NOCS map to 6D pose parameters in the Pose Network. This allows the gradients to backprop through the network end-to-end. We will explain the architecture and objective details in the following subsections.

4.1 Framework Details

Input data pre-processing. Given an RGBD image, we first utilize the off-the-shelf Mask R-CNN [12] model to infer the segmentation mask for each object instance (See supplementary material for segmentation examples). We crop an object of interest from the RGB image and the corresponding depth map. The pixels in the depth map are back-projected as an observed point cloud. We denote the observation of an object instance as $\mathrm{X} \in \mathbb{R}^{H \times W \times 3}$ for RGB channels and $\mathrm{P} \in \mathbb{R}^{N_\mathrm{p} \times 3}$ for the point clouds, where W, H stands for cropped image resolution and N_p is the number of sampled points.

Feature extraction. We employ the PSPNet [56] to extract the image feature as f_x (Fig. 3 green box), and a PointNet [36] to extract the point cloud geometry feature as f_p (Fig. 3 blue box). Meanwhile, for each category, we first define a category-level mesh prior $\mathrm{M} \in \mathbb{R}^{N_\mathrm{v} \times 3}$ to represent objects belong to a specific category, where N_v is the number of vertices of the pre-defined mesh. We also use the PointNet to extract the shape prior feature as f_cate (Fig. 3 purple box).

4.1.1 Pose Network.

Unlike instance-level 6D object pose estimation, where the CAD model of a specific instance is always given as the reference, the Pose Network is excepted to work well even without any instance models. To achieve this goal, we leverage the NOCS representation proposed in [48]. The architecture of the Pose Network is shown as the bottom part of Fig. 3. The Pose Network takes the RGB features f_x extracted using PSPNet and the point cloud features f_p extracted using PointNet as inputs. The two different modalities are combined as RGBD features using the dense fusion approach proposed in [45]. Specifically, the geometry feature f_p^i of the i th point is concatenated with its corresponding color feature f_x^i and it is fed into a Graph Convolution Network (GCN) [27] (Fig. 3 yellow box) to obtain the point-wise RGBD feature, denote as f_rgbd^i.

Meanwhile, recall that we utilize a PointNet to obtain the point-wise geometry feature of the category-level shape prior f_cate (on the Shape Network branch). The shape prior contains the prior knowledge of the category, *i.e.*coarse canonical shape and pose, but it does not align with the RGBD feature perfectly. To use the information of categorical shape prior and address the misalignment, we first apply a max-pooling operation on point dimension to obtain a global representation, then concatenate it with every point-wise RGBD feature, denoted as f_nocs^i, which can be used to estimate the NOCS coordinate. As the size and geometry of real-world object can be quite diverse, we introduce an **implicit function** $\Phi_\mathrm{nocs}(\cdot)$ to deal with objects of the arbitrary shape (Fig. 3 gray box with MLP). Specifically, the $\Phi_\mathrm{nocs}(\cdot)$ is implemented as a MLP with the point position and the corresponding feature as inputs, and predicts its NOCS coordinate as,

$$\Phi_\mathrm{nocs}(f_\mathrm{nocs}^i, \mathrm{p}^i \in \mathrm{P}) = \mathrm{p}_\mathrm{nocs}^i, \quad \forall i := 1 \ldots N_\mathrm{p}, \tag{1}$$

where P is the set of input point clouds and N_p is the number of point clouds. The output is the NOCS coordinate $\mathrm{P}_\mathrm{nocs}^i$ for each point.

The NOCS map also explicitly reflects the geometric shape information of objects, similar properties are also shown with representations in [47]. Thus, the 6D pose, *i.e.*, rotation (**R**), translation (**T**), and scale (**S**) can be inferred from the NOCS map via a simple neural network. In detail, we directly take the RGBD feature f_rgbd and concatenate it with the **intermediate results of NOCS coordinates**. This concatenated feature f_pose will be used to predict the object pose **R**, **T** and **S** by a 3-layer ConvNet (Fig. 3 gray box with CNN). Using a neural network to connect the NOCS map and the 6D pose parameters makes the whole process differentiable, which is essential for training with the silhouette matching loss. More details of the network architecture can be found in the supplementary materials.

Inference for 6D pose. For 6D object pose estimation, only the NOCS map output of Pose Network is required for inference. The object pose can be computed by solving the Umeyama algorithm [43]

https://doi.org/10.52202/068431-1992

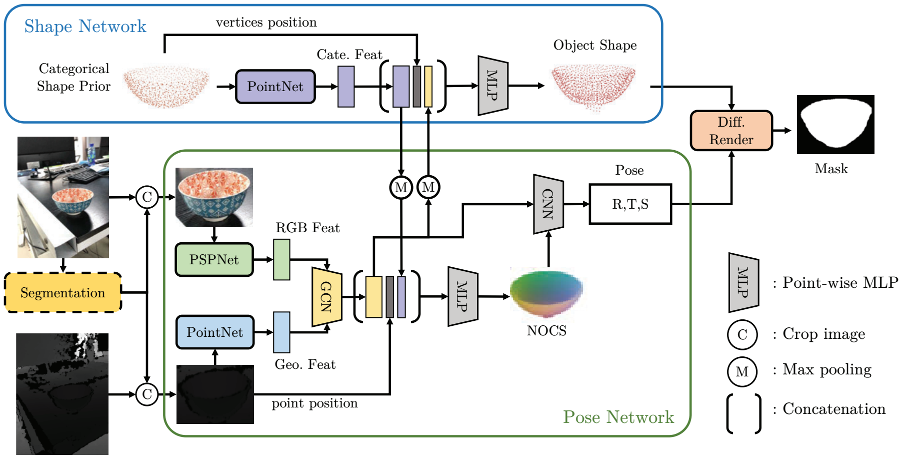

Figure 3: Overview of the proposed method. Given the input image and depth map, RePoNet estimates the object pose, NOCS map, and shape simultaneously via Pose Network and Shape Network. These two networks are bridged via the differentiable rendering module. By comparing the predicted binary mask with the input foreground mask, RePoNet can effectively leverage the real-world data without any annotations.

with the NOCS map. The 6D pose parameters $\mathbf{R}$, $\mathbf{T}$ and $\mathbf{S}$ are only used to perform differentiable rendering during training. The reason is that the directly estimated pose parameters from conv module is accurate when fitting the training data, but not as accurate as using NOCS map when generalizing to novel test instances.

4.1.2 Shape Network.

The Shape Network (top part of Fig. 3) aims to reconstruct the 3D shape of the input object from the given shape prior via mesh deformation. The major usage of this network is for performing differentiable rendering to provide training signals. The Shape Network is only used to estimate the object scale during inference time.

Given the categorical shape prior representation f_{cate}, the Shape Network borrows the point-wise RGBD feature of the input object f_{rgbd} from Pose Network and concatenate them together, denoted as f_{shape}. Specifically, we max-pool the features f_{rgbd} on point dimension and repeat it back with the number of shape prior points for concatenation with f_{cate}. We then predict the per-vertices deformation for the input mesh given f_{shape}. The deformation prediction network $\Phi_{\text{deform}}(\cdot)$ is also an implicit function taking f_{shape} and mesh vertices positions M as the inputs, and outputs the 3D deformation as,

$$\Phi_{\text{deform}}(f^i_{\text{shape}}, \mathrm{m}^i \in \mathrm{M}) = \mathrm{m}^i_{\text{delta}}, \quad \forall i := 1 \dots N_{\text{v}}, \tag{2}$$

where $\mathrm{m}^i_{\text{delta}}$ represents the deformation for each vertex, N_{v} is the number of the vertices, and $\mathrm{M}_{\text{delta}} = \{\mathrm{m}^i_{\text{delta}}\}_{i=1}^{N_{\text{v}}}$. Then the estimated mesh vertices of the given object is $\mathrm{M}_{\text{deform}} = \mathrm{M} + \mathrm{M}_{\text{delta}}$.

4.1.3 Differentiable rendering.

To obtain the supervision from 2D segmentation, we perform differentiable rendering using the outputs from the Shape Network and the Pose Network. Specifically, we feed the estimated object shape $\mathrm{M}_{\text{deform}}$ and estimated object pose $(\mathbf{R}, \mathbf{T}, \mathbf{S})$ into a differentiable rendering module [28] to generate the binary mask of the given object, as shown in Fig. 3. Given the rendering output binary mask, we can compare it against the Mask R-CNN segmentations and supervise the training of both pose network and shape network without additional 3D annotations.

4.2 Learning Objectives

We define multiple objectives for training the RePoNet, including the loss for 6D pose parameters, NOCS regression loss, shape reconstruction loss using synthetic ground-truths for supervision, and a silhouette matching loss which can be applied to both synthetic data and real data without 6D pose annotations. We introduce each loss as the following.

Disentangled 6D Pose Loss. Inspired by [24, 47, 39], we implement disentangled pose loss via individually supervising the rotation $\mathbf{R}$, translation $\mathbf{T}$ and scale $\mathbf{S}$. Instead of directly computing

https://doi.org/10.52202/068431-1992

parametric distances based on rotation matrix, we employ the variant of Point-Matching loss[1] [24, 20]. Additionally, we decouple the translation into the 2D location (o_x, o_y) of the 3D centroid projection on the image plane and the object's distance to camera t_z. (o_x, o_y) can be approximated as the bounding box center of the given object (c_x, c_y). Given the camera intrinsics K, the translation can be calculated via Eq. 3,

$$\mathbf{T} = K^{-1} t_z [o_x, o_y, 1]^T \tag{3}$$

Therefore, given the ground-truth pose annotations $(\mathbf{R}^*, \mathbf{T}^*, \mathbf{S}^*)$, the objective function of 6D pose is formulated as Eq. 4,

$$\mathcal{L}_{\text{pose}} = \mathcal{L}_{\mathbf{R}} + \mathcal{L}_{\text{center}} + \mathcal{L}_z + \mathcal{L}_{\mathbf{S}} \tag{4}$$

Thereby,

$$\mathcal{L}_{\mathbf{R}} = \underset{\mathbf{x}\in \mathrm{M}_{\mathrm{CAD}}}{\text{avg}} \; \|\mathbf{R}\mathbf{x} - \mathbf{R}^*\mathbf{x}\|_2 \quad \mathcal{L}_{\text{center}} = \|o_x - o_x^*, o_y - o_y^* * \|_1 \quad \mathcal{L}_z = \|t_z - t_z^*\|_1 \quad \mathcal{L}_{\mathbf{S}} = \|\mathbf{S} - \mathbf{S}^*\|_1 \tag{5}$$

where $\mathrm{M}_{\mathrm{CAD}}$ is the ground-truth CAD model of the given object and o_x^*, o_y^*, t_z^* can be computed via Eq. 3.

NOCS Regression Loss. To guarantee the pose estimation accuracy, we enforce the predicted NOCS coordinates P_{nocs} to be closer to the ground-truth ones $\mathrm{P}^*_{\text{nocs}}$. We use smooth-L1 loss [11],

$$\mathcal{L}_{\text{nocs}} = \sum_{x,y} \begin{cases} 0.5(x-y)^2/\beta, \text{if } |x-y| < \beta \\ |x-y| - 0.5 * \beta, \text{otherwise} \end{cases} \tag{6}$$

where $x \in \mathrm{P^*}_{\text{nocs}}, y \in \mathrm{P}_{\text{nocs}}$

Reconstruction Loss. Given the ground-truth CAD model $\mathrm{M}_{\mathrm{CAD}}$, the shape estimation is supervised by minimizing Chamfer Distance(CD) between $\mathrm{M}_{\text{deform}}$ and $\mathrm{M}_{\mathrm{CAD}}$ via Eq. 7,

$$\mathcal{L}_{\text{recon}} = \sum_{x\in \mathrm{M}_{\text{deform}}} \min_{y\in \mathrm{M}_{\mathrm{CAD}}} \|x-y\|_2^2 + \sum_{y\in \mathrm{M}_{\mathrm{CAD}}} \min_{x\in \mathrm{M}_{\text{deform}}} \|x-y\|_2^2 \tag{7}$$

Silhouette Matching Loss. The above three objective functions, $\mathcal{L}_{\text{pose}}$, $\mathcal{L}_{\text{nocs}}$ and $\mathcal{L}_{\text{recon}}$, require the ground-truth information and cannot work with the real-world data without annotations. Thus, we further implement a silhouette matching objective function based on input foreground segmentation and the rendering results. We employ the negative IOU loss [54] to measure the difference between two masks and denote it as $\mathcal{L}_{\text{mask}}$.

Therefore, for the **fully-supervised training** with annotated synthetic data, the total loss function is described in Eq. 8, where the λs are balance parameters.

$$\mathcal{L}_{\text{sup}} = \lambda_1 \mathcal{L}_{\text{pose}} + \lambda_2 \mathcal{L}_{\text{nocs}} + \lambda_3 \mathcal{L}_{\text{recon}} + \lambda_4 \mathcal{L}_{\text{mask}} \tag{8}$$

Meanwhile, under the **semi-supervised setting** with both annotated synthetic data and unlabeled data, the total loss function is then formulated as:

$$\mathcal{L}_{\text{semi}} = \mathbb{1}_{d\in \mathbf{D}_{\text{syn}}} (\lambda_1 \mathcal{L}_{\text{pose}} + \lambda_2 \mathcal{L}_{\text{nocs}} + \lambda_3 \mathcal{L}_{\text{recon}}) + \lambda_4 \mathcal{L}_{\text{mask}} \tag{9}$$

where $\mathbb{1}_{d\in \mathbf{D}_{\text{syn}}}$ is the indicator function.

5 Experiments

5.1 Implementation Details

Semi-supervised setting. We use the training data of CAMERA25 [48] along with the corresponding annotations and jointly train the model with images of REAL275 [48] or Wild6D without any 6D pose annotations. After cropping the object from the RGBD image, we first resize it to 192 × 192 and then randomly sample 1,024 points from both color image and depth map. To obtain the categorical shape prior, we choose a CAD model per category from the CAMERA25 training set manually and reduce its number of vertices to 1,024 as well. More configurations of RePoNet have been specified in supplementary materials. We adopt Adam [19] to optimize our model with the initial learning rate of 0.0001. The learning rate is halved every 10 epochs until convergence. We empirically set the balance parameters $\lambda_1, \lambda_2, \lambda_3$ and λ_4 to 0.2, 2.0, 5.0 and 0.2, respectively.

Fully-supervised setting. We share the same architecture and configurations with semi-supervised training but utilize all annotations of CAMERA25 [48] and REAL275 [48].

[1] We predict the quaternion representation of rotation matrix and follow the strategy in [50, 47] to deal with the symmetry of the object.

https://doi.org/10.52202/068431-1992

NOCS map	Implicit function	$IOU_{0.5}$	$IOU_{0.75}$	5 degree 5cm	10 degree 5cm	$IOU_{0.5}$	$IOU_{0.75}$	5 degree 5cm	10 degree 5cm
		Fully supervised				**Semi-supervised**			
		77.0	55.7	35.4	63.2	71.7	47.9	22.1	47.5
✓		79.1	59.1	36.9	65.3	74.6	49.9	29.7	57.1
	✓	79.2	60.0	39.8	66.6	72.2	47.7	24.3	51.6
✓	✓	**81.1**	**63.7**	**40.4**	**68.8**	**76.0**	**52.2**	**33.9**	**63.0**

Table 2: **Ablation on Implicit function and intermediate NOCS map.** We evaluate on REAL275 and the best results are highlighted in bold. w./w.o the nocs map refers to whether using the NOCS map as an intermediate result for pose regression. w./w.o. the implicit function stands for whether including the point/mesh coordinates in NOCS regression and shape reconstruction.

silhouette loss	pose loss	$IOU_{0.5}$	$IOU_{0.75}$	5 degree 5cm	10 degree 5cm	$IOU_{0.5}$	$IOU_{0.75}$	5 degree 5cm	10 degree 5cm
		Fully supervised				**Semi-supervised**			
✓		78.1	61.9	39.7	65.2	74.8	47.1	27.9	59.7
	✓	76.7	58.2	30.7	59.7	73.1	46.0	18.7	47.9
✓	✓	**81.1**	**63.7**	**40.4**	**68.8**	**76.0**	**52.2**	**33.9**	**63.0**

Table 3: **Ablation study on pose regression loss and silhouette matching loss.** We evaluate on REAL275 and the best results are highlighted in bold.

5.2 Ablation Study

Implicit function and NOCS map. We first validate the effectiveness of two components in our approach: implicit function and intermediate NOCS map as described in Sec. 4.1.1. We evaluate the performance under fully-supervised setting and semi-supervised setting on REAL275 [48] and report the results in Table 2. Under the fully-supervised setting, by using the implicit function, the performance on $IOU_{0.5}$ and 5 degree, 5cm is improved by 2.2% and 4.4%, respectively. A similar improvement can also be observed for the model with an intermediate NOCS map. The best performance is achieved when combining the implicit function and NOCS map together. And the same conclusion can also be obtained under the semi-supervised setting as shown in Table 2.

Pose Loss and Silhouette Matching Loss. Besides the network architecture, we also show the effectiveness of the disentangled 6D pose loss and silhouette matching loss. The performance of the model w./w.o. the pose regression loss and w./w.o. the silhouette matching loss are listed in Table 3. Clearly, both pose loss and silhouette matching loss can enhance the object pose estimation performance under both fully-supervised and semi-supervised settings. For instance, the silhouette matching loss improves the $IOU_{0.75}$ and 10 degree 5cm by 5.5% and 9.1% compared with the model without it under the fully-supervised setting. This significant improvement shows the rendering module can effectively encourage communication between two networks, thereby boosting the pose estimation performance.

Leveraging unlabeled real data. Additionally, we report the performance of the proposed semi-supervised approach with training data from different sources in Table 4. As the baseline model, we first train it only with the synthetic data from CAMERA25 in a fully-supervised manner. Then, by using the unlabeled Wild3D RGBD videos, the performance is significantly improved by 6.8% and 5.4% on $IOU_{0.75}$ and 5 degree, 2cm. Similarly, the performance gains can be observed when adopting the REAL275 data without its annotations. Furthermore, with both REAL275 and Wild3D data, our model outperforms the baseline model by a large margin and approaches the fully-supervised model trained on REAL275 which can be served as the upper bound. More ablation studies about the amount of unlabeled real data used can be found in the supplementary materials.

Amount of unlabeled real data. We analyze the effect of using different fractions of unlabeled real data used during semi-supervised learning. We uniformly sample every 10% fraction of collected Wild6D training data for semi-supervised learning and evaluate the performance on REAL275 and Wild6D testing sets. As shown in Fig. 4, with more real data used during training, the object pose estimation performance is getting better.

5.3 Comparison with State-of-the-art Methods

Performance on REAL275. We split all existing methods into two groups by whether using full annotations of real data and compare their performance with RePoNet under both fully-supervised setting and semi-supervised setting, denoted as “RePoNet-sup" and “RePoNet-semi" respectively. As listed in Table 5, RePoNet-sup can achieve competitive results among all existing approaches under

Methods	Training Data			IOU$_{0.25}$	IOU$_{0.5}$	IOU$_{0.75}$	5 degree	5 degree	10 degree	10 degree
	CAMERA25	REAL275	Wild6D				2cm	5cm	2cm	5cm
RePoNet-semi	✓			83.8	73.5	45.7	18.2	21.8	35.6	47.9
	✓		✓	85.0	74.3	52.5	23.6	26.1	45.2	53.8
	✓	✓		85.8	76.9	49.2	29.1	31.3	48.5	56.8
	✓	✓	✓	86.0	78.5	51.8	30.8	36.0	48.7	59.7
RePoNet-sup	✓	✓		86.6	83.0	63.2	31.1	37.0	51.6	64.6

Table 4: **Ablation study of proposed semi-supervised method with different training data.** We evaluate the performance on REAL275 test set. “RePoNet-semi" and “RePoNet-sup" represent for the model trained w./w.o the 3D annotations of NOCS REAL275 respectively. Here we conduct experiments on five categories sharing by both REAL275 and Wild6D, *i.e.*bottle, bowl, mug, laptop and camera.

Methods	IOU$_{0.5}$	5 degree 2cm	5 degree 5cm	10 degree 5cm
NOCS [48]	78.0	7.2	10.0	25.2
CASS [3]	77.7	–	23.5	58.0
Shape-Prior [41]	77.3	19.3	21.4	54.1
FS-Net [6]	**92.2**	–	28.2	60.8
DualPoseNet [25]	79.8	29.3	35.9	66.8
SGPA [4]	80.1	**35.9**	39.6	70.7
GPV-Pose [10]	83.0	32.0	**42.9**	**73.3**
RePoNet-sup	81.1	35.1	40.4	68.8
CPS++ [29] w/ ICP	72.8	–	25.2	≤ 58.6
SSC-6D [35] w/ ICP	72.7	28.6	33.4	62.9
CPPF [53]	26.4	–	16.9	44.9
UDA-COPE [21]	**82.6**	30.4	**34.8**	**66.0**
RePoNet-semi	76.0	**30.7**	33.9	63.0

Table 5: **Comparison of our approach with the SOTA methods on REAL275**. Note that all SOTA methods listed in the top part are trained with full annotations of NOCS dataset including CAMERA25 and REAL275, while the bottom part is methods without using any real annotations. The best results are highlighted in bold.

the fully-supervised setting. Note our method is complimentary to the techniques proposed in SGPA, and our key contribution and focus lies on the semi-supervised counterpart for generalization. As shown in the bottom part of Table 5, the RePoNet-semi outperforms CPS++ [29] by a large margin under the semi-supervised setting [2]. Moreover, compared with the existing fully-supervised methods[3], RePoNet-semi achieves a comparable or even better performance without any 6D pose annotations. Additionally, our approach can perform shape reconstruction, but it's not our goal and we report its performance in the supplementary material.

Performance on Wild6D. Finally, we evaluate the performance of RePoNet on the Wild6D testing set as reported in Table 6. For some existing works, we directly test their pre-trained models trained on CAMERA75 along with REAL275. Since FS-Net [6] does not release the model, we cannot experiment on it. It can be observed that the pre-trained models cannot generalize to Wild6D due to the limited diversity of real data during training. For example, the performance of Shape-Prior [41] on three pose estimation metrics are all lower than 15%. On the other hand, the RePoNet with semi-supervised learning on Wild6D achieves 29.5% and 34.4% on 5 degree, 2cm and 5 degree, 5cm, which is 10× better than Shape-Prior [41]. This significant improvement shows the better generalization ability of RePoNet. Also, the superior performance of RePoNet-semi over RePoNet-syn shows our proposed semi-supervised training can effectively leverage the in-the-wild data. We also show the qualitative results on Wild6D test set in Fig. 5.

Methods	IOU$_{0.5}$	5 degree 2cm	5 degree 5cm	10 degree 5cm
CASS [3]	1.04	0	0	0
Shape-Prior [41]	32.5	2.6	3.5	13.9
DualPoseNet [25]	70.0	17.8	22.8	36.5
GPV-Pose [10]	67.8	14.1	21.5	41.1
RePoNet-syn	66.7	26.0	30.8	40.3
RePoNet-semi	**70.3**	**29.5**	**34.4**	**42.5**

Table 6: **Comparison of our approach with the SOTA methods on Wild6D.** “RePoNet-syn" is the model trained only on CAMERA75, while “RePoNet-semi" stands for the RePoNet trained on CAMERA75 and Wild6D. The best results are highlighted in bold.

[2] In Table 5, CPS++ only reports the performance under 10 degree, 10 cm.

[3] FS-Net and GPV-Pose are treated as the fully-supervised one, since it requires all real data and annotations.

https://doi.org/10.52202/068431-1992

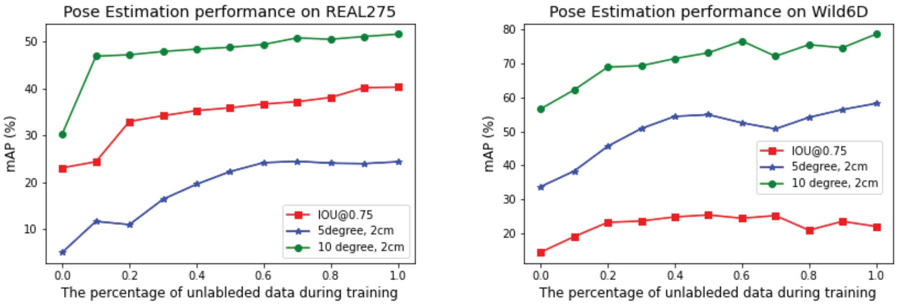

Figure 4: **Ablation study on semi-supervised training with different number of unlabeled real data**. Here, we only show the performance on bottle and evaluate it on both REAL275 and Wild6D dataset.

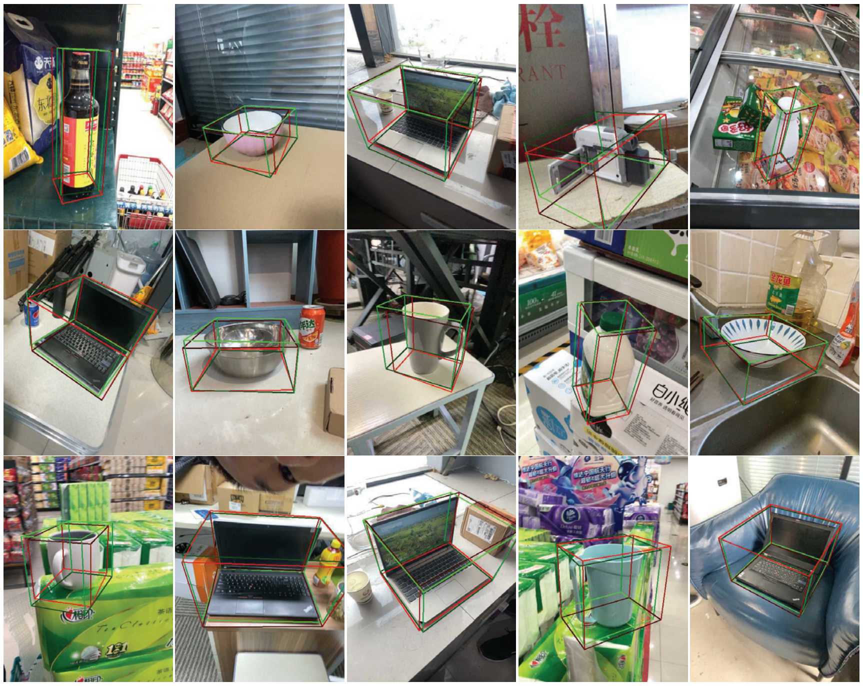

Figure 5: **Visualization Results on Wild6D test set**. Red 3D bounding boxes denote the ground truth, and the green boxes are estimation results via our proposed method.

6 Discussion

Conclusion. In this work, we consider the problem of generalizing 6D pose estimation to in-the-wild objects. Most of existing work are restricted in constrained environments due to limited number of annotated data. In an effort to resolve this limitation, we collect *Wild6D*, a new RGBD dataset with diverse instances and backgrounds. Instead of annotating them, we also propose RePoNet, a network that can leverage both the synthetic data and real-world RGBD data. Without using any 3D annotations on real-world data, RePoNet outperforms state-of-the-art methods on existing datasets and Wild6D test set by a large margin.

Limitations. Our proposed RePoNet may be hard to generalize to unseen categories. This is because RePoNet depends on the categorical mesh prior which is pre-defined for each category leading to the learnt deformation model fixed to a specific category. Alternative way to address this problem is to represent object shape via a deformation of a unit sphere to remove the category-level prior. Our dataset is currently limited to 5 categories to align with previous work for evaluation. We will extend it to more categories in the future.

Acknowledgement. This work was supported, in part, by grants from DARPA LwLL, NSF CCF-2112665 (TILOS), NSF 1730158 CI-New: Cognitive Hardware and Software Ecosystem Community Infrastructure (CHASE-CI), NSF ACI-1541349 CC*DNI Pacific Research Platform, and gifts from Meta, Google, Qualcomm.

References

[1] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7822–7831, 2021. 3, 4

[2] Frank Biocca. Virtual reality technology: A tutorial. *Journal of communication*, 42(4):23–72, 1992. 1

[3] Dengsheng Chen, Jun Li, Zheng Wang, and Kai Xu. Learning canonical shape space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11973–11982, 2020. 1, 3, 9

[4] Kai Chen and Qi Dou. Sgpa: Structure-guided prior adaptation for category-level 6d object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2773–2782, 2021. 9

[5] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4233–4242, 2020. 1

[6] Wei Chen, Xi Jia, Hyung Jin Chang, Jinming Duan, Linlin Shen, and Ales Leonardis. Fs-net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1581–1590, 2021. 1, 3, 9

[7] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. *Advances in Neural Information Processing Systems*, 32:9609–9619, 2019. 3

[8] Xu Chen, Zijian Dong, Jie Song, Andreas Geiger, and Otmar Hilliges. Category level object pose estimation via neural analysis-by-synthesis. In *European Conference on Computer Vision*, pages 139–156. Springer, 2020. 3

[9] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016. 3

[10] Yan Di, Ruida Zhang, Zhiqiang Lou, Fabian Manhardt, Xiangyang Ji, Nassir Navab, and Federico Tombari. Gpv-pose: Category-level object pose estimation via geometry-guided point-wise voting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6781–6791, 2022. 9

[11] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 7

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 4, 5

[13] 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

[14] Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 1

[15] Tingbo Hou, Adel Ahmadyan, Liangkai Zhang, Jianing Wei, and Matthias Grundmann. Mobilepose: Real-time pose estimation for unseen objects with weak shape supervision. *arXiv preprint arXiv:2003.03522*, 2020. 3

[16] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M Kitani. Repose: Fast 6d object pose refinement via deep texture rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3303–3312, 2021. 3

[17] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018. 3

[18] Hiroharu Kato and Tatsuya Harada. Learning view priors for single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9778–9787, 2019. 3

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

https://doi.org/10.52202/068431-1992

[20] 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. 3, 7

[21] Taeyeop Lee, Byeong-Uk Lee, Inkyu Shin, Jaesung Choe, Ukcheol Shin, In So Kweon, and Kuk-Jin Yoon. Uda-cope: Unsupervised domain adaptation for category-level object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14891–14900, 2022. 9

[22] Chi Li, Jin Bai, and Gregory D Hager. A unified framework for multi-view multi-class object pose estimation. In *Proceedings of the european conference on computer vision (eccv)*, pages 254–269, 2018. 1

[23] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3d reconstruction via semantic consistency. In *European Conference on Computer Vision*, pages 677–693. Springer, 2020. 3

[24] 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. 3, 6, 7

[25] Jiehong Lin, Zewei Wei, Zhihao Li, Songcen Xu, Kui Jia, and Yuanqing Li. Dualposenet: Category-level 6d object pose and size estimation using dual pose network with refined learning of pose consistency. *arXiv preprint arXiv:2103.06526*, 2021. 3, 9

[26] Yunzhi Lin, Jonathan Tremblay, Stephen Tyree, Patricio A Vela, and Stan Birchfield. Single-stage keypoint-based category-level object pose estimation from an rgb image. *arXiv preprint arXiv:2109.06161*, 2021. 3

[27] Zhi-Hao Lin, Sheng-Yu Huang, and Yu-Chiang Frank Wang. Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1800–1809, 2020. 5

[28] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019. 3, 4, 6

[29] Fabian Manhardt, Gu Wang, Benjamin Busam, Manuel Nickel, Sven Meier, Luca Minciullo, Xiangyang Ji, and Nassir Navab. Cps++: Improving class-level 6d pose and shape estimation from monocular images with self-supervised learning. *arXiv preprint arXiv:2003.05848*, 2020. 1, 3, 9

[30] Eitan Marder-Eppstein. Project tango. In *ACM SIGGRAPH 2016 Real-Time Live!*, pages 25–25. 2016. 1

[31] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2901–2910, 2019. 1

[32] 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. 1

[33] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. In *Proceedings of the IEEE international conference on computer vision*, pages 143–152, 2017. 4

[34] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: 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

[35] Wanli Peng, Jianhang Yan, Hongtao Wen, and Yi Sun. Self-supervised category-level 6d object pose estimation with deep implicit shape representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2082–2090, 2022. 9

[36] 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. 5

[37] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021. 3

https://doi.org/10.52202/068431-1992

[38] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 3

[39] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1991–1999, 2019. 6

[40] Juil Sock, Guillermo Garcia-Hernando, Anil Armagan, and Tae-Kyun Kim. Introducing pose consistency and warp-alignment for self-supervised 6d object pose estimation in color images. In *2020 International Conference on 3D Vision (3DV)*, pages 291–300. IEEE, 2020. 3

[41] Meng Tian, Marcelo H Ang, and Gim Hee Lee. Shape prior deformation for categorical 6d object pose and size estimation. In *European Conference on Computer Vision*, pages 530–546. Springer, 2020. 1, 3, 9

[42] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018. 1

[43] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991. 2, 3, 5

[44] Angtian Wang, Adam Kortylewski, and Alan Yuille. Nemo: Neural mesh models of contrastive features for robust 3d pose estimation. *arXiv preprint arXiv:2101.12378*, 2021. 3

[45] 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. 5

[46] Gu Wang, Fabian Manhardt, Jianzhun Shao, Xiangyang Ji, Nassir Navab, and Federico Tombari. Self6d: Self-supervised monocular 6d object pose estimation. In *European Conference on Computer Vision*, pages 108–125. Springer, 2020. 3

[47] 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*, pages 16611–16621, 2021. 5, 6, 7

[48] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 1, 2, 3, 4, 5, 7, 8, 9

[49] Bowen Wen, Chaitanya Mitash, Sruthi Soorian, Andrew Kimmel, Avishai Sintov, and Kostas E Bekris. Robust, occlusion-aware pose estimation for objects grasped by adaptive hands. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6210–6217. IEEE, 2020. 1

[50] 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, 7

[51] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2020. 4

[52] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. *arXiv preprint arXiv:2012.05877*, 2020. 3

[53] Yang You, Ruoxi Shi, Weiming Wang, and Cewu Lu. Cppf: Towards robust category-level 9d pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6866–6875, 2022. 9

[54] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520, 2016. 7

[55] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1941–1950, 2019. 3

[56] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 5

https://doi.org/10.52202/068431-1992

[57] Menglong Zhu, Konstantinos G Derpanis, Yinfei Yang, Samarth Brahmbhatt, Mabel Zhang, Cody Phillips, Matthieu Lecce, and Kostas Daniilidis. Single image 3d object detection and pose estimation for grasping. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3936–3943. IEEE, 2014. 1

https://doi.org/10.52202/068431-1992

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplementary
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See section 5
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplementary
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See supplementary
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

https://doi.org/10.52202/068431-1992