# REDE: End-to-end Object 6D Pose Robust Estimation Using Differentiable Outliers Elimination

Weitong Hua[1], Zhongxiang Zhou[1], Jun Wu[1], Huang Huang[2], Yue Wang[1], Rong Xiong[1]

*Abstract*—Object 6D pose estimation is a fundamental task in many applications. Conventional methods solve the task by detecting and matching the keypoints, then estimating the pose. Recent efforts bringing deep learning into the problem mainly overcome the vulnerability of conventional methods to environmental variation due to the hand-crafted feature design. However, these methods cannot achieve end-to-end learning and good interpretability at the same time. In this paper, we propose REDE, a novel end-to-end object pose estimator using RGB-D data, which utilizes network for keypoint regression, and a differentiable geometric pose estimator for pose error back-propagation. Besides, to achieve better robustness when outlier keypoint prediction occurs, we further propose a differentiable outliers elimination method that regresses the candidate result and the confidence simultaneously. Via confidence weighted aggregation of multiple candidates, we can reduce the effect from the outliers in the final estimation. Finally, following the conventional method, we apply a learnable refinement process to further improve the estimation. The experimental results on three benchmark datasets show that REDE slightly outperforms the state-of-the-art approaches and is more robust to object occlusion. Our code is available at *https://github.com/HuaWeitong/REDE*.

*Index Terms*—Deep Learning for Visual Perception, RGB-D Perception, Pose Estimation
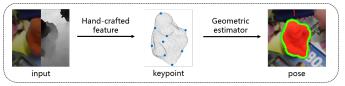
## I. INTRODUCTION

THE task of object 6D pose estimation is to predict the 3D rotation and 3D translation of the object in the current scene with respect to the world coordinates fixed on the object. It is very important in many applications, such as augmented reality [1], automatic driving [2], [3] and robot grasping [4], [5]. Conventionally, a popular method for object pose estimation is built upon hand-crafted keypoint detection and feature matching. The pose is then estimated utilizing the point correspondences between the current image and the object model [6]–[8]. The main weakness of this pipeline is that the feature design is not learnable to fit the data, leading to difficulties on pose estimation for featureless objects. More recently, with the progress of deep learning, network based pose estimation becomes more popular. There

(a) Conventional line.



(b) Direct regression line.



(c) Keypoint regression line.



(d) REDE.

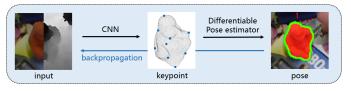Fig. 1. Illustration comparing conventional methods (a), direct regression methods (b), keypoint regression methods (c) and REDE (d). All trainable components are highlighted in blue. REDE integrates the end-to-end learning in *direct regression*, into the *keypoint regression*.

are mainly two lines of works. In the first line, named as *direct regression*, the pose estimation is learned by a fully connected network as a regression task from feature to pose [9]–[11], thus achieving the end-to-end learning. However, due to the highly nonlinearity structure in rotation, such methods have weaker generalization. To overcome this challenge, the other line of works, named as *keypoint regression*, only utilize network for keypoint detection and regression. Then, following the conventional pipeline, the pose is then calculated based on the keypoint matching, leaving the nonlinear rotation to geometric estimator [12]–[14]. Obviously, this line of works loses the end-to-end training of the network.

At the perspective of sensor modalities, with the progress in quality and lower cost, RGB-D sensor becomes a new preference for object pose estimation. With the aid of depth, the keypoints can be represented in 3D, thus overcoming the

information degeneration in 3D-2D configuration. For 3D-3D correspondences, a closed form solution can be achieved, avoiding more degenerated cases [15], [16]. In addition, depth also provide important discriminative cues for occlusion and cluttered scenes, which is common in robotic objects manipulation [17].

In this paper, we set to build an RGB-D object pose estimator, which integrates the end-to-end learning in *direct regression*, into the *keypoint regression*, so that geometric estimator is employed without losing the end-to-end training. The main ideas of these different lines of works are illustrated in Fig. 1. Moreover, we introduce the differentiable outliers elimination into the geometric estimator, so that initial pose solution can be estimated more robustly. Then, the pose is further improved via the iterative refinement [11]. Note that this pipeline imitates the conventional pipeline more tightly, which robustly estimates the initial pose using robust solver like RANSAC [18], and refines the pose using nonlinear optimization [11]. Our psychology is to leave the feature construction to learning as it is conventionally achieved by empirical design, but keep geometry to calculus and optimization as it is already guaranteed by theory.

In summary, this work has the following contributions:

- A differentiable outliers elimination mechanism is designed by softly aggregating the keypoints positions and candidate poses from minimal solvers bank, so that the robustness of pose estimation is improved.
- An end-to-end object pose estimation network is proposed by differentiating the geometric estimator, so that gradients can be back-propagated from final pose loss to the keypoint prediction.
- The performance of the proposed method is evaluated on three large public benchmark datasets, YCB-Video, LineMOD and Occlusion LineMOD, showing state-of-the-art performance with only pose annotation.

## II. RELATED WORK

### A. Template matching methods

While object CAD model is available in this task, some traditional methods select the most appropriate feature embedding of object model compared with the scene data and calculate the pose transformation. PPF [6] employs point cloud and proposes a characteristic point pair feature. During inference phase, the ppf features of scene points are matched with the ppf feature of model points and cast votes to pose. [19] proposes novel sampling and voting strategies to avoid sensor noise and background clutter. Hinterstoisser [7], [8] calculates contour gradient vectors from color image and surface normal vectors from depth image respectively to compose multimodal features for template matching. AAE [20] proposes an augmented auto-encoder to encode only pose information of models from various views, then compares the similarity with real image embedding. CosyPose [21] carries out the matching and joint optimization of objects in the whole scene based on the observation of multiple views. Template matching based methods can easily estimate the general result with exact CAD model of the object. But a great quantity of matching is time-consuming, and the pose obtained is the discrete result, which leads to less accuracy.

### B. Learning based regression methods

Due to the strong fitting ability of CNN, there are also some methods which encode features from RGB or RGB-D data and directly classify or regress pose. SSD-6D [22] extends SSD [23] framework and classifies decomposed pose in anchor-based way. PoseCNN [9] votes for center translation and regresses rotation respectively from RGB image. MCN [24] decouples translation and rotation, then divides them into several bins for classification. DenseFusion [11] proposes a dense fusion strategy which fuses color embedding and geometric embedding in point-wise way, then regresses pose directly. The question of classification is still the imprecision caused by discretization. As for direct regression, learning ability of the network is limited because of the nonlinearity of rotation space. Moreover, there is no special treatment for occlusion in this way.

### C. 2D-3D correspondence methods

Inspired by keypoint detection based on RGB image, many methods convert this issue to 2D projection of keypoint location, and then calculate pose by PnP algorithm to obtain 2D-3D correspondence. BB8 [25] employs CNN to detect eight corners of 3D bounding box. YOLO-6D [12] and 3D-SSD [26] extend YOLO and SSD respectively to predict 3D bounding box corners. However, the corners of bounding box are not on the surface of the object, which leads to large localization errors. Therefore, PVNet [13] employs the farthest point sampling (FPS) algorithm to select more representative keypoints. Besides, PVNet predicts the direction vector from each pixel pointing to the projection point and employs RANSAC voting strategy to locate the projection points, which can acquire more robust results. [27] further proposes DPVLoss to constrain the distance between point and vector, which makes vector prediction more robust. More recent methods develop dense prediction of 2D-3D correspondence and estimate pose by PnP-RANSAC. Pix2Pose [28] generates 3D coordinate mapping image from 2D pixels of input image based on GAN training. DPOD [14] predicts dense 2D-3D mapping between RGB image and model with UV texture map projection. EPOS [29] predicts 2D-3D correspondence based on fragments and estimates pose based on a variant of PnP-RANSAC. These methods based on RGB image need two-stage inference, and rely on the quality of point prediction. DSAC [30] proposes differentiable ransac for camera localization. [31] makes efforts for combining the two stage into one for end-to-end training, but the pose is inferred by MLP, which still cannot overcome the shortcoming of direct regression lines.

## III. END-TO-END ROBUST POSE ESTIMATION

An overview of REDE is shown in Fig. 2. In the first stage, the RGB-D frame is fed into the network to generate point-wise point-to-keypoint offsets and confidences. In the second stage, we calculate the 3D keypoints by aggregating the
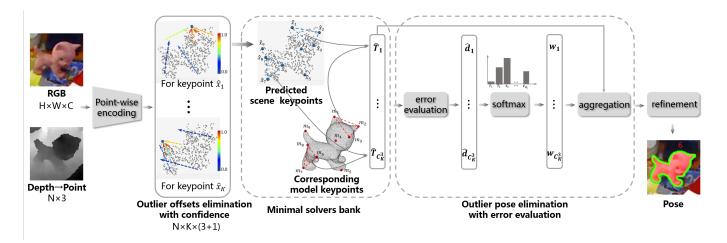
Fig. 2. Overview of REDE: The point-wise point-to-keypoint offsets are predicted from the segmented RGB-D frame. We predict the corresponding confidences meanwhile for the robust aggregation of 3D keypoints. Then multiple poses are estimated by exhaustively enumerating every 3 keypoints among all candidate keypoints to build minimal solvers bank. Next, the pose is estimated robustly by weighted aggregation with differentiable outliers elimination and final refinement.

offsets. In the third stage, we estimate multiple poses and their confidences by exhaustively enumerating every 3 keypoints among all candidate keypoints to build minimal solvers bank. The confidence is a mapping of the residue between scene points and model points transformed using the corresponding estimated pose, leading to a robust aggregation to outlier pose. At last, the initial result is further refined for better accuracy.

### A. RGB-D Feature Encoding

Before pose estimation, we need to segment region of interest in image and point cloud. Since segmentation is not the subject of our study, we use the existing segmentation results of [9], or ground truth masks as previous works [15]. Then we follow [11] to extract point-wise feature embedding: the RGB image segment is encoded into color embedding using PSPNet [32], and the point cloud segment is encoded into geometric embedding using PointNet [33]. To build the point-wise features, the geometric embedding is concatenated with its corresponding color embedding. Then the global feature vector is generated by average pooling layer and further concatenates to each point feature. After these processes, we build dense point-wise feature embedding containing both color and geometric information, as well as local and global information.

### B. Robust 3D Keypoint Prediction

We take 3D keypoint as a mediator to estimate pose. As [13], we employ the farthest point sampling (FPS) algorithm to sample 3D keypoints $\{m_k\}_{k=1}^K$ from the CAD model of each object. Then it is required to locate corresponding 3D keypoints in the current RGB-D image, which in previous works is achieved by regressing the relative direction from each point to the keypoint, followed by an aggregation of all estimations. This method is considered to be more stable, and learn the spatial structure characteristics of point cloud more effectively. However, this process is not differentiable, thus [13] has no back-propagation after keypoint regression.

**Offset regression:** To solve this problem, thanks to the RGB-D information, we predict the offset from each point to the keypoint, which is also followed an aggregation of all estimations. In this way, the aggregation is simply a closed form of averaging, thus the keypoint regression becomes differentiable. In detail, for the keypoint of scene $\{x_k\}_{k=1}^K$, the offset from input scene point $\{s_i\}_{i=1}^N$ should be:

$$v_{k,i} = x_k - s_i \tag{1}$$

We utilize the smooth L1 error in $x$, $y$ and $z$ directions as the loss term. The loss term for offset vector prediction is defined as the following equations:

$$\mathcal{L}oss_{vec} = \sum_{k=1}^K \sum_{i=1}^N L(\Delta v_{k,i}|_x) + L(\Delta v_{k,i}|_y) + L(\Delta v_{k,i}|_z) \tag{2}$$

$$\Delta v_{k,i} = \hat{v}_{k,i} - v_{k,i} \tag{3}$$

$$L(x) = \begin{cases} 0.5x^2, & if(|x| < 1) \\ |x| - 0.5, & otherwise \end{cases} \tag{4}$$

where $\hat{v}_{k,i}$ is the predicted offset, $\cdot|_x$, $\cdot|_y$ and $\cdot|_z$ are the $x$, $y$ and $z$ component of $\cdot$.

**Outlier offset elimination:** Since predicted offsets with large errors are inevitable, which are outliers in estimation, the simple averaging for offset prediction is not robust. Therefore, we design a differentiable outliers elimination method for robust keypoint prediction. In addition to the offset, we also regress confidence for each offset, denoted as $c_{k,i}$. With the confidence $c_{k,i}$, the predicted offset $\hat{v}_{k,i}$ is softly aggregated to get a position of the keypoint $\hat{x}_k$ as

$$\hat{x}_k = \sum_{i=1}^N c_{k,i}(s_i + \hat{v}_{k,i}) \tag{5}$$

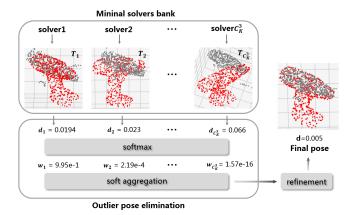In this way, confidence can be learned using pose loss.

Fig. 3. A case of our pose estimator using minimal solvers bank and differentiable outliers elimination mechanism, in which "driller" is under severe occlusion. For each candidate pose from minimal solvers bank, the residue between scene points and model points transformed using the corresponding estimated pose is calculated and mapped to confidence. Outlier pose derived using incorrect keypoint is assigned to low confidence, leading to a robust aggregation.

### C. Robust Differentiable Pose Estimator

Given the predicted 3D keypoints $\{\hat{x}_k\}_{k=1}^K$, we can solve the object pose based on the keypoints $\{m_k\}_{k=1}^K$ in the object coordinate. The solver is built upon the optimization of the relative distance between the predicted keypoints, and the transformed model keypoints using the object pose. It can be written as

$$\hat{R}, \hat{t} = \arg\min_{R,t} \sum_{k=1}^K ||(R \cdot m_k + t) - \hat{x}_k||^2 \quad (6)$$

where $R$ and $t$ form the 6D object pose. Fortunately, this optimization problem can be solved by SVD in a closed form. Therefore, it can be easily embedded in our network without losing end-to-end training.

**Minimal solvers bank:** Since the RGB-D data can only be captured in one view, there can be some unobservable keypoints due to occlusion. Therefore, it is highly possible that the network can not precisely predict them. As a result, the potential outliers may degenerate the performance seriously. A simple idea is to follow the outlier elimination in keypoint prediction that we can assign a weight for each keypoint and employ the weighted least square algorithm. But we find that the performance becomes worse, probably because the network is prone to give up many keypoints. DSAC [30] proposes differentiable ransac in the field of camera localization. But it is used for dense points, which has to adopt stochastic sampling strategy.

To make the estimator more robust, we propose minimal solvers bank. For every 3 keypoints, we can solve a pose as a candidate using the minimal version of (6). Thanks to the small number of keypoints, we can permutate all possibilities to generate a bank of $C_K^3$ minimal solvers. This module can be evaluated efficiently based on GPU and closed form.

**Outlier pose elimination:** Given the resultant set of candidate pose $\{\hat{T}_i\}_{i=1}^{C_K^3}$, which can be split into rotation $\{\hat{R}_i\}_{i=1}^{C_K^3}$ and translation $\{\hat{t}_i\}_{i=1}^{C_K^3}$, we need to aggregate them to gen-

erate a robust estimation. We still use confidence to softly average the candidates. However, different from the outlier offset elimination, we first calculate the error distance between scene points $\{s_j\}_{j=1}^N$ and their nearest neighbor among the transformed model points $\{p_j\}_{j=1}^N$ using the corresponding candidate pose, say $\hat{T}_i$

$$\hat{d}_i = \sum_{j=1}^N ||(\hat{R}_i \cdot p_{N(s_j)} + \hat{t}_i) - s_j|| \quad (7)$$

where $p_{N(s_j)}$ is the nearest neighbor of the scene point $s_j$ among the transformed model points. Then the confidence $w_i$ is predicted as a monotonic mapping of the error distance. Repeating the process for all the candidate poses, we have a confidence vector for all candidates in $\{\hat{T}_i\}_{i=1}^{C_K^3}$.

To normalize the weights, we apply softmax to each error $d_i$ to derive the final weights $\{w_i\}_{i=1}^{C_K^3}$ as

$$w_i = \frac{e^{-\frac{\hat{d}_i}{\lambda}}}{\sum_{j=1}^{C_K^3} e^{-\frac{\hat{d}_j}{\lambda}}} \quad (8)$$

where $\lambda$ is a temperature coefficient. With the weights, we can softly aggregate the translation by weighted averaging

$$\hat{t} = \sum_{i=1}^{C_K^3} w_i \cdot \hat{t}_i \quad (9)$$

For rotation, the aggregation is slightly different due to the nonlinear structure of the rotation space. We first transform the rotation matrix $\{\hat{R}_i\}_{i=1}^{C_K^3}$ into quaternion $\{\hat{q}_i\}_{i=1}^{C_K^3}$ for aggregation since averaging the rotation matrices can break the $SO(3)$ constraints. After that, the quaternions are aggregated and normalized to get the unit vector, which is depicted as:

$$q' = \sum_{i=1}^{C_K^3} w_i \cdot \hat{q}_i, \quad \hat{q} = \frac{q'}{||q'||} \quad (10)$$

**Differentiable pose estimation:** We arrive at the estimated pose $\{\hat{q}, \hat{t}\}$, which is robust to the outliers caused by incorrect offset regression. Since the whole robust pose estimator is differentiable, we can train the keypoint prediction network with error signal from not only the keypoint ground truth, but also the pose ground truth. The latter loss term is defined as

$$\mathcal{L}oss_{pose} = ||\hat{t} - t||_2 + \alpha||\hat{R} \cdot R^T - I||_F \quad (11)$$

where $\alpha$ is a balancing parameter, $R$ and $t$ are the ground truth pose, $\hat{R}$ is coverted from $\hat{q}$.

In total, the network is trained by a joint loss

$$\mathcal{L}oss = \mathcal{L}oss_{vec} + \beta \cdot \mathcal{L}oss_{pose} \quad (12)$$

where $\beta$ is a trade-off parameter between the two terms. Finally, following the conventional pipeline that nonlinearly refines the initial value derived from the robust estimator, we also further iteratively optimize the pose via iterative refinement in [11]. A heavily occluded case of our pose estimator using minimal solvers bank and differentiable outliers elimination mechanism is illustrated in Fig. 3.

TABLE I
ADD-S PERFORMANCE ON YCB-VIDEO DATASET.

| | with GT mask | | | | | | | | with PoseCNN mask | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DenseFusion [11] | | P²GNet [34] | | PVN3D (w/o semantics) [15] | | REDE | | PoseCNN [9] | | DenseFusion [11] | | Tian et al. [35] | | REDE | |
| refine | √ | | √ | | | | √ | | √ | | √ | | | | √ | |
| metric | AUC | <2cm | AUC | <2cm | AUC | <2cm | AUC | <2cm | AUC | <2cm | AUC | <2cm | AUC | <2cm | AUC | <2cm |
| 002 | 96.2 | 100.0 | - | - | - | - | 95.4 | 100.0 | 95.8 | 100.0 | 96.4 | 100.0 | 93.9 | - | 95.1 | 100.0 |
| 003 | 95.3 | 100.0 | - | - | - | - | 96.4 | 100.0 | 92.7 | 91.6 | 95.5 | 99.5 | 92.9 | - | 96.3 | 99.7 |
| 004 | 97.9 | 100.0 | - | - | - | - | 98.0 | 100.0 | 98.2 | 100.0 | 97.5 | 100.0 | 95.4 | - | 97.4 | 100.0 |
| 005 | 94.3 | 96.9 | - | - | - | - | 96.2 | 98.4 | 94.5 | 96.9 | 94.6 | 96.9 | 93.3 | - | 96.9 | 100.0 |
| 006 | 97.7 | 100.0 | - | - | - | - | 98.0 | 100.0 | 98.6 | 100.0 | 97.2 | 100.0 | 95.4 | - | 96.7 | 100.0 |
| 007 | 96.7 | 100.0 | - | - | - | - | 96.9 | 100.0 | 97.1 | 100.0 | 96.6 | 100.0 | 94.9 | - | 96.6 | 100.0 |
| 008 | 97.3 | 100.0 | - | - | - | - | 97.8 | 100.0 | 97.9 | 100.0 | 96.5 | 100.0 | 94.0 | - | 96.4 | 100.0 |
| 009 | 98.4 | 100.0 | - | - | - | - | 98.8 | 100.0 | 98.8 | 100.0 | 98.1 | 100.0 | 97.6 | - | 97.8 | 100.0 |
| 010 | 90.2 | 92.3 | - | - | - | - | 92.1 | 94.5 | 92.7 | 93.6 | 91.3 | 93.1 | 90.6 | - | 92.0 | 94.2 |
| 011 | 96.2 | 99.7 | - | - | - | - | 97.7 | 100.0 | 97.1 | 99.7 | 96.6 | 100.0 | 91.7 | - | 97.0 | 99.7 |
| 019 | 97.5 | 100.0 | - | - | - | - | 98.1 | 100.0 | 97.8 | 100.0 | 97.1 | 100.0 | 93.1 | - | 97.5 | 100.0 |
| 021 | 96.4 | 100.0 | - | - | - | - | 96.7 | 100.0 | 96.9 | 99.4 | 95.8 | 100.0 | 93.4 | - | 94.2 | 100.0 |
| 024 | 88.9 | 87.4 | - | - | - | - | 96.6 | 99.5 | 81.0 | 54.9 | 88.2 | 98.8 | 92.9 | - | 96.7 | 99.3 |
| 025 | 97.0 | 100.0 | - | - | - | - | 97.5 | 100.0 | 95.0 | 99.8 | 97.1 | 100.0 | 96.1 | - | 97.0 | 100.0 |
| 035 | 97.1 | 100.0 | - | - | - | - | 97.9 | 100.0 | 98.2 | 99.6 | 96.0 | 98.7 | 93.3 | - | 97.0 | 99.6 |
| 036 | 94.1 | 100.0 | - | - | - | - | 93.8 | 100.0 | 87.6 | 80.2 | 89.7 | 94.6 | 87.6 | - | 91.0 | 98.3 |
| 037 | 93.2 | 100.0 | - | - | - | - | 93.5 | 96.7 | 91.7 | 95.6 | 95.2 | 100.0 | 95.7 | - | 94.5 | 100.0 |
| 040 | 97.5 | 100.0 | - | - | - | - | 98.1 | 100.0 | 97.2 | 99.7 | 97.5 | 100.0 | 95.6 | - | 97.8 | 100.0 |
| 051 | 89.7 | 98.0 | - | - | - | - | 96.9 | 100.0 | 75.2 | 74.9 | 72.9 | 79.2 | 75.4 | - | 77.3 | 80.7 |
| 052 | 77.4 | 80.5 | - | - | - | - | 96.2 | 99.9 | 64.4 | 48.8 | 69.8 | 76.3 | 73.0 | - | 85.9 | 82.0 |
| 061 | 91.5 | 100.0 | - | - | - | - | 95.5 | 100.0 | 97.2 | 100.0 | 92.5 | 100.0 | 94.2 | - | 94.6 | 100.0 |
| MEAN | 94.2 | 97.8 | 94.2 | 97.8 | 94.8 | - | **96.6** | **99.5** | 93.0 | 93.2 | 93.1 | 96.8 | 91.8 | - | **94.5** | **97.8** |

## IV. EXPERIMENTS

To validate the proposed method, we report the performance by comparing it with the state-of-the-art methods on three datasets, YCB-Video dataset [9], LineMOD dataset [7] and Occlusion LineMOD dataset [36].

### A. Datasets

YCB-Video dataset contains 21 objects with various textrues from YCB objects. There are 92 RGB-D videos in which 80 videos are used for training and 2949 keyframes from the rest 12 videos are used for testing. Besides, 80000 synthetic images are released for training. There are many scenes of stacking objects with partial occlusion.

LineMOD dataset is a widely used benchmark for object 6D pose estimation task. This dataset contains 13 objects totally. We follow prior learning-based works [11], [15] to split training and testing data. There are about 180 training images and 1000 testing images for each object. 10000 images using the "Cut and Paste" strategy are further synthesized for training as [13].

Occlusion LineMOD dataset further annotates data with serious occlusion in LineMOD dataset. It contains 8 objects and 1214 images. All images are used for evaluation with model trained on LineMOD dataset. The main challenge on this dataset is severe occlusion, especially for small target.

### B. Metrics

The most commonly used metrics for object pose estimation are ADD [8] and ADD-S [9]. ADD metric is defined as the average Euclidean distance between model points transformed with the predicted and the ground truth pose respectively:

$$ADD = \frac{1}{N} \sum_{i=1}^{N} ||(R \cdot p_i + t) - (\hat{R} \cdot p_i + \hat{t})|| \quad (13)$$

where $N$ is the number of model points $\{p_i\}_{i=1}^{N}$, $R$ and $t$ are the rotation and translation of ground truth pose, $\hat{R}$ and $\hat{t}$ are the rotation and translation of predicted pose.

ADD-S metric is designed for symmetric object and calculates the average distance with the closest point:

$$ADD\text{-}S = \frac{1}{N} \sum_{i=1}^{N} \min_{j \in [1,N]} ||(R \cdot p_j + t) - (\hat{R} \cdot p_i + \hat{t})|| \quad (14)$$

For YCB-Video dataset, we report the results of two metrics with ADD-S, the same as [11]. The first is the correct percentage in all data which ADD-S is smaller than 2cm. The second is the area under the ADD-S curve (AUC), which is obtained by varying the distance threshold in evaluation.

For LineMOD dataset and Occlusion LineMOD dataset, we use ADD(-S) metric following prior works [8]. For non-symmetric objects, we use ADD metric. For symmetric objects (eggbox and glue in two datasets), due to the ambiguity of pose, we use ADD-S metric. We regard the evaluation result as accurate if ADD(-S) is less than 10% of the object model's diameter.

### C. Analysis

We explore the effectiveness of our differentiable outliers elimination mechanism for keypoint location and pose estimation in this part.

TABLE II
ADD(-S) PERFORMANCE ON LINEMOD DATASET.

| | RGB | | | RGB-D[1] | | | | |
|---|---|---|---|---|---|---|---|---|
| | PVNet [13] | DPOD [14] | DPVL [27] | PointFusion [3] | DenseFusion [11] | $P^2$GNet [34] | Tian et al. [35] | REDE |
| ape | 43.6 | 87.7 | 69.1 | 70.4 | 92.3 | 92.9 | 85.0 | 95.6 |
| benchvise | 99.9 | 98.5 | 100.0 | 80.7 | 93.2 | 98.2 | 95.5 | 99.4 |
| cam | 86.9 | 96.1 | 94.1 | 60.8 | 94.4 | 97.0 | 91.3 | 99.6 |
| can | 95.5 | 99.7 | 98.5 | 61.1 | 93.1 | 97.4 | 95.2 | 99.5 |
| cat | 79.3 | 94.7 | 83.1 | 79.1 | 96.5 | 98.1 | 93.6 | 99.5 |
| driller | 96.4 | 98.8 | 99.0 | 47.3 | 87.0 | 97.0 | 82.6 | 99.3 |
| duck | 52.6 | 86.3 | 63.5 | 63.0 | 92.3 | 95.2 | 88.1 | 97.0 |
| eggbox | 99.2 | 99.9 | 100.0 | 99.9 | 99.8 | 100.0 | 99.9 | 100.0 |
| glue | 95.7 | 96.8 | 98.0 | 99.3 | 100.0 | 100.0 | 99.6 | 99.9 |
| holepuncher | 81.9 | 87.7 | 88.2 | 71.8 | 86.9 | 97.9 | 92.6 | 98.6 |
| iron | 98.9 | 100.0 | 99.9 | 83.2 | 97.0 | 98.2 | 95.9 | 99.3 |
| lamp | 99.3 | 96.8 | 99.8 | 62.3 | 95.3 | 97.7 | 94.4 | 99.3 |
| phone | 92.4 | 94.7 | 96.4 | 78.8 | 92.8 | 96.7 | 93.6 | 99.3 |
| average | 86.3 | 95.2 | 91.5 | 73.3 | 94.3 | 97.4 | 92.9 | **98.9** |

[1] All the RGB-D based methods listed here use PoseCNN mask.

TABLE III
ADD(-S) PERFORMANCE ON OCCLUSION LINEMOD DATASET.

| | PVNet [13] | DPOD [14] | DPVL [27] | PVN3D [15] | REDE[1] |
|---|---|---|---|---|---|
| ape | 15.8 | - | 19.2 | 33.9 | 53.1 |
| can | 63.3 | - | 69.8 | 88.6 | 88.5 |
| cat | 16.7 | - | 21.1 | 39.1 | 35.9 |
| driller | 65.7 | - | 71.6 | 78.4 | 77.8 |
| duck | 25.2 | - | 34.3 | 41.9 | 46.2 |
| eggbox | 50.2 | - | 47.3 | 80.9 | 71.8 |
| glue | 49.6 | - | 39.7 | 68.1 | 75.0 |
| holepuncher | 39.7 | - | 45.3 | 74.7 | 75.5 |
| average | 40.8 | 47.3 | 43.5 | 63.2 | **65.4** |

[1] With the same mask as PVNet [13].

TABLE IV
ABLATION STUDIES FOR LOSS.

| offset loss | pose loss | ADD(-S)<0.1d |
|---|---|---|
| √ | | 59.6 |
| √ | √ | 65.4 |

[1] With the same mask as PVNet [13] on Occlusion LineMOD dataset.

**Confidence visualization:** The confidences of offsets are visualized to observe the effect of differentiable outliers elimination mechanism in keypoint location. As shown in Fig. 4, we project points on RGB image with varying colors to show confidences. The confidence increases as color changes from blue to red. We can see from "040_larger_marker" that the offset estimated by the point close to the keypoint tends to be assigned to higher confidence, which is in line with our intuition. Besides, the textured areas of "007_tuna_fish_can" and "005_tomato_soup_can" also have higher confidence. More interestingly, the confidences of the edge points in

TABLE V
ABLATION STUDIES FOR DIFFERENTIABLE OUTLIERS ELIMINATION.

| DOE for keypoint | DOE for pose | ADD-S AUC | ADD(-S) AUC |
|---|---|---|---|
| | | 92.3 | 85.4 |
| √ | | 92.7 | 86.6 |
| √ | √ | 94.3 | 89.5 |

[1] With PoseCNN mask on YCB-Video dataset.



040_large_marker    007_tuna_fish_can    005_tomato_soup_can    024_bowl

Fig. 4. Visualization for confidence of offsets. Red color indicates high confidence and blue indicates low confidence. The ground truth keypoint and predicted keypoint are marked with white and black triangle dots respectively.

"024_bowl" are higher, but the confidences of the points around the keypoint are low instead. It can be concluded that the prediction of the points with small offsets and obvious characteristics is more accurate, so the confidence is higher.

**Ablation studies:** We conduct two ablation studies to verify the effects of different losses and differentiable outliers elimination(DOE). The addition of pose loss achieves 5.8% improvement on Occlusion LineMOD dataset (see Table IV), which verify the importance of end-to-end training. As for our differentiable outliers elimination, table V summarizes the results of ablation studies on YCB-Video dataset. Here we report the results using PoseCNN mask without ICP. For ADD-S metric, Our AUC is 0.4% higher with DOE for keypoint location and 2.0% higher with DOE for pose estimation. For ADD(-S) metric, Our AUC is 1.2% higher with DOE for keypoint location and 4.1% higher with DOE for pose estimation.

*D. Results on Benchmark Dataset*

**Evaluation on YCB-Video Dataset.** Cause the large clamp and extra large clamp in YCB-Video dataset are the same models with two different scales, the segmentation network with RGB image as the only input can not distinguish them. In order not to be affected by the ambiguous segmentation results from PoseCNN [9], we also employ ground truth mask to verify our pose estimation network. The evaluation results for this two kind of masks are both listed in Table I. All methods listed in the table are based on RGB-D data and use only pose related supervision. REDE outperforms others
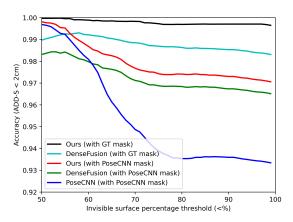
Fig. 5. Performance curve for different invisible surface percentage. REDE has more robust performance under heavy occlusion especially when not affected by bad segmentation results.

on both two metrics and two kind of masks. The numerical values we report in table are the results after ICP refinement. The ADD-S AUC of our method using ground truth mask is 96.2% without ICP and 95.6% without refinement, which also surpasses 94.8% in PVN3D [15]. Fig. 6 displays some visualization results. It can be observed that compared with DenseFusion [11], our method can accurately estimate the pose of objects, especially in some hard cases with occlusion.

To verify our robustness towards occlusion, we also draw accuracy curve under increasing levels of occlusion on YCB-Video dataset. Following DenseFusion [11], levels of occlusion are measured by calculating the invisible surface percentage of model points in the image frame. The accuracy of ADD-S smaller than 2cm curve is shown in Fig. 5, it can be seen that our performance under occlusion is more stable, especially in the case of accurate segmentation results.

**Evaluation on LineMOD and Occlusion LineMOD Dataset.** Our quantitative evaluation results of the pose estimation experiments on the LineMOD dataset are reported in Table II. We have to clarify that all the RGB-D based methods listed here use the same segmentation masks released by PoseCNN. The ADD(-S) metric of our method outperforms all other approaches. Especially, the performance of prior works on the small objects such as "ape", "cat" and "duck" are poor due to the few available pixels in the image frame. But we can handle them very well which is benefit from end-to-end learning and differentible outliers elimination.

As for Occlusion LineMOD dataset with many hard cases, we use the same masks as PVNet [13]. We report the quantitative results in Table III, where PVN3D [15] is also based on RGB-D. We achieve accuracy of 65.4% which outperforms other recent methods. This proves that our differentiable outliers elimination mechanism works well for partial occlusion, and the robustness of the estimation is improved.

### E. Runtime

On GTX 2080 Ti GPU, REDE takes 0.03 seconds for pose estimation and refinement. With 0.03 seconds for prior instance segmentation, the overall runtime is about 17 FPS

on the LineMOD dataset, which is promising for real-time applications.

## V. CONCLUSION

We present REDE, an end-to-end object 6D pose robust estimation method based on keypoint regression in this paper. We integrate the end-to-end learning into keypoint regression, so as to get better supervision. In addition, we design a differentiable outliers elimination mechanism for both keypoint location and pose estimation, which solves the problem of keypoint prediction deviation probably caused by occlusion. Experiments show that REDE outperforms the state-of-the-art methods in several datasets and is especially robust for outliers.

In the future, we will consider some studies combined with other fields. First, multi-task learning will be conducted together with segmentation. Second, we will make use of the advantages of end-to-end manner for self-supervised learning. Finally, conducting transfer learning from synthetic data only is also worth to be studied further.

## REFERENCES

[1] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: a hands-on survey," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 12, pp. 2633–2651, 2015.

[2] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[3] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3d bounding box estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 244–253.

[4] M. Zhu, K. G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce, and K. Daniilidis, "Single image 3d object detection and pose estimation for grasping," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3936–3943.

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

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

[7] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of textureless objects in heavily cluttered scenes," in *international conference on computer vision*. IEEE, 2011, pp. 858–865.

[8] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian conference on computer vision*. Springer, 2012, pp. 548–562.

[9] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Robotics: Science and Systems*, 2018.

[10] T.-T. Do, M. Cai, T. Pham, and I. Reid, "Deep-6dpose: Recovering 6d object pose from a single rgb image," *arXiv preprint arXiv:1802.10367*, 2018.

[11] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3343–3352.

[12] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 292–301.

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

**Ground Truth**

**DenseFusion**

**Our REDE**

Fig. 6. Some visualization results on YCB-Video dataset. The point cloud of all objects are transformed with the predicted pose and projected to the RGB image using different colors. Compared with DenseFusion, our REDE estimates the poses more accurately.

[14] S. Zakharov, I. Shugurov, and S. Ilic, "Dpod: 6d pose object detector and refiner," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1941–1950.

[15] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 632–11 641.

[16] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.

[17] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[18] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[19] S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige, "Going further with point pair features," in *European conference on computer vision*. Springer, 2016, pp. 834–848.

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

[21] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Cosypose: Consistent multi-view multi-object 6d pose estimation," in *European Conference on Computer Vision*. Springer, 2020, pp. 574–591.

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

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[24] C. Li, J. Bai, and G. D. Hager, "A unified framework for multi-view multi-class object pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 254–269.

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

[26] Q. Luo, H. Ma, L. Tang, Y. Wang, and R. Xiong, "3d-ssd: Learning hierarchical features from rgb-d images for amodal 3d object detection," *Neurocomputing*, vol. 378, pp. 364–374, 2020.

[27] X. Yu, Z. Zhuang, P. Koniusz, and H. Li, "6dof object pose estimation via differentiable proxy voting loss," *arXiv preprint arXiv:2002.03923*, 2020.

[28] K. Park, T. Patten, and M. Vincze, "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7668–7677.

[29] T. Hodan, D. Barath, and J. Matas, "Epos: Estimating 6d pose of objects with symmetries," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 703–11 712.

[30] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac-differentiable ransac for camera localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6684–6692.

[31] Y. Hu, P. Fua, W. Wang, and M. Salzmann, "Single-stage 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2930–2939.

[32] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[33] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[34] P. Yu, Y. Rao, J. Lu, and J. Zhou, "P²gnet: Pose-guided point cloud generating networks for 6-dof object pose estimation," *arXiv preprint arXiv:1912.09316*, 2019.

[35] M. Tian, L. Pan, M. H. Ang Jr, and G. H. Lee, "Robust 6d object pose estimation by learning rgb-d features," in *International Conference on Robotics and Automation (ICRA)*, 2020.

[36] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6d object pose estimation using 3d object coordinates," in *European conference on computer vision*. Springer, 2014, pp. 536–551.

TABLE A1
ABLATION STUDY FOR END-TO-END TRAINING ON OCCLUSION
LINEMOD DATASET.

| | w/o end-to-end | | | end-to-end |
|---|---|---|---|---|
| | DPVL (3D version) | PVNet (3D version) | PVNet (3D version with RANSAC) | REDE |
| estimator | SVD 3D-3D | SVD 3D-3D | RANSAC 3D-3D | DOE |
| ape | 55.7 | 51.6 | 58.3 | 53.1 |
| can | 73.2 | 75.6 | 74.9 | 88.5 |
| cat | 16.9 | 28.7 | 30.2 | 35.9 |
| driller | 68.5 | 66.9 | 73.3 | 77.8 |
| duck | 39.3 | 36.7 | 35.2 | 46.2 |
| eggbox | 39.8 | 47.1 | 46.8 | 71.8 |
| glue | 67.7 | 71.9 | 74.5 | 75.0 |
| holepuncher | 50.1 | 45.7 | 53.8 | 75.5 |
| average | 51.1 | 52.6 | 55.5 | 65.4 |

TABLE A2
ABLATION STUDIES FOR DIFFERENTIABLE OUTLIERS ELIMINATION USING
GROUND TRUTH MASK.

| DOE for keypoint | DOE for pose | ADD-S AUC | ADD(-S) AUC |
|---|---|---|---|
| | | 94.9 | 89.3 |
| √ | | 95.2 | 90.2 |
| √ | √ | 96.2 | 92.2 |

## APPENDIX A
## IMPLEMENTATION DETAILS

During implementation, center point and 8 points selected by FPS algorithm are picked up as keypoints following PVNet [13]. The number of input scene points sampled is 1000 for YCB-Video dataset and 500 for LineMOD dataset. Data augmentation strategies such as random illumination variation are applied to enhance robustness towards brightness and background. 10000 images using the "Cut and Paste" strategy are further synthesized for training on LineMOD dataset as [13]. The refinement network proposed in DenseFusion [11] is further employed to iteratively optimize the pose. For YCB-Video dataset, we also employ the ICP algorithm [16] to improve the performance. The learning rate is set to 1e-4 in pose estimaiton network and 3e-5 in refinement network. The refine margin of ADD(-S) metric is set to 0.013 in YCB-Video

TABLE A3
ABLATION STUDIES FOR DIFFERENTIABLE OUTLIERS ELIMINATION USING
POSECNN MASK.

| DOE for keypoint | DOE for pose | ADD-S AUC | ADD(-S) AUC |
|---|---|---|---|
| | | 92.3 | 85.4 |
| √ | | 92.7 | 86.6 |
| √ | √ | 94.3 | 89.5 |

TABLE A4
ABLATION STUDIES FOR DIFFERENTIABLE OUTLIERS ELIMINATION USING
POSECNN MASKS WITH ARTIFICIAL NOISES.

| DOE for keypoint | DOE for pose | ADD-S AUC | ADD(-S) AUC |
|---|---|---|---|
| | | 92.1 | 84.8 |
| √ | | 92.5 | 86.5 |
| √ | √ | 94.0 | 88.9 |



Fig. A1. Some visualization results on LineMOD and Occlusion LineMOD dataset. In the pictures, the pose result is shown by projecting the model using the estimated pose. For the convenience of viewing, we also show the contour of the model projection.

dataset and 0.01 in LineMOD dataset. The trade-off parameter $\alpha$ is set to 0.01 and $\beta$ is set to 0.1.

## APPENDIX B
### ANYLASIS FOR END-TO-END TRAINING

In ablation studies for end-to-end training, we conduct three extra experiments without end-to-end manner for comparision on Occlusion LineMOD dataset. The first experiment extends DPVL [27] to 3D and employs SVD 3D-3D estimator. The second and third experiments extend PVNet [13] to 3D and the second experiment also employs SVD 3D-3D estimator. The third experiment employs RANSAC 3D-3D estimator instead without end-to-end training. Instead of RANSAC, our REDE designs differentiable outliers elimination(DOE) as estimator to obtain deterministic rather than stochastic results, while it is derivable to achieve end-to-end manner. These three results are all worse than REDE (see Table A1). Our REDE also outperforms other methods which also employ 3D keypoints but without end-to-end training like PVN3D [15], which can prove the importance of end-to-end training.

## APPENDIX C
### ANYLASIS FOR DIFFERENTIABLE OUTLIERS ELIMINATION

In ablation studies for differentiable outliers elimination, we conduct three experiments on YCB-Video dataset using different kinds of masks: ground truth mask, PoseCNN mask and PoseCNN mask with artificial noises. For the last mask, we erode PoseCNN mask several times to add artificial noises. Table A2, Table A3 and Table A4 report the results respectively. The results for all three types of masks show the effectiveness of our differentiable outliers elimination module. Besides, the improvements using two kinds of inaccurate masks is indeed more obvious, which further proves the robustness of our method to outliers.

## APPENDIX D
### MORE RESULTS

Some visualization results on LineMOD and Occlusion LineMOD dataset are shown in Fig. A1.