

Article

Few-Shot 6D Object Pose Estimation via Decoupled Rotation and Translation with Viewpoint Encoding

Lei Lu ¹, Peng Cao ¹, Wei Pan ^{2,*} , Zhilong Su ^{3,4}, Haojun Zhang ⁵, Wangxing Zheng ⁵, Ge Gao ⁶ and Peng Li ^{1,*} 

¹ Institute for Complexity Science, Henan University of Technology, Zhengzhou 450001, China; lulei@haut.edu.cn (L.L.); caopeng@stu.haut.edu.cn (P.C.)

² Department of R&D, OPT Machine Vision Tech Co., Ltd., Dongguan 523860, China

³ Shanghai Institute of Applied Mathematics and Mechanics, School of Mechanics and Engineering Science, Shanghai Key Laboratory of Mechanics in Energy Engineering, Shanghai University, Shanghai 200444, China

⁴ Shaoxing Research Institute of Shanghai University, Shaoxing 312074, China

⁵ College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China

⁶ Mech-Mind Robotics Technologies Ltd., Beijing 110000, China

* Correspondence: vpan@foxmail.com (W.P.); lipeng@haut.edu.cn (P.L.)

Abstract

Estimating 6D object pose from monocular RGB images remains a critical yet data-intensive challenge in computer vision. In this work, we propose a novel few-shot 6D pose estimation framework that explicitly decouples rotation and translation estimation, significantly reducing dependence on large-scale annotated real-world data. Our method employs a viewpoint encoder trained solely on synthetic data to generate a codebook for rotation retrieval, complemented by an in-plane rotation regression module. For translation, we adopt a geometry-aware regression network based on dense 2D–3D correspondences. Experimental results on LINEMOD, LM-O, and YCB-V datasets demonstrate that our approach achieves state-of-the-art performance (97.6%, 65.3%, and 65.9% ADD(-S), respectively), using only 600 real images per object—cutting real data requirements by 80% compared to typical fully-supervised 6D pose estimation methods. These findings highlight the effectiveness and generalization ability of our method under limited supervision.

Keywords: few-shot; 6D pose estimation; convolutional neural network; viewpoint encoding; deep learning

1. Introduction

6D pose estimation refers to determining the pose of an object in three-dimensional space, including its location (3D translation) and orientation (3D rotation), and is one of the core research directions in computer vision. This technology plays a critical role in various applications such as robotic grasping in industrial settings [1–3], autonomous driving [4,5], and augmented reality [6,7]. However, depth data is susceptible to environmental noise, occlusion, and hardware cost constraints, and it also faces limitations when dealing with reflective or transparent objects. These challenges motivate the exploration of new 6D object pose estimation solutions based solely on monocular RGB data.

With the advances in deep learning, a growing number of novel 6D pose estimation methods based on monocular RGB data have emerged, gradually overcoming the limitations of data-driven approaches and, in some cases, even surpassing methods that rely on depth data [8,9]. Current mainstream approaches can be categorized into three paradigms: (1) estimating 6D pose by establishing 2D–3D correspondences through precise 2D key-point extraction [10–12]; (2) template-based methods that compare input images against



Academic Editor: Danilo Pau

Received: 19 December 2025

Revised: 25 January 2026

Accepted: 27 January 2026

Published: 28 January 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

multi-view object templates for pose estimation [13,14]; and (3) end-to-end frameworks that directly regress 6D pose parameters [15,16]. Although these monocular RGB-based methods have achieved notable progress, their practical deployment remains constrained by the deep learning models' heavy reliance on large-scale annotated datasets.

To address this challenge, we propose a decoupled pose estimation network tailored for few-shot scenarios, which independently estimates rotation and translation components of the 6D pose. Here, “few-shot” is defined relative to standard 6D pose estimation benchmarks: while fully supervised methods typically require thousands or even tens of thousands of real annotated images, our method achieves competitive performance using only a small number of real images (e.g., approximately 600 images per object, corresponding to less than 5–20% of full training sets such as YCB-Video). We acknowledge that the term ‘few-shot’ differs from its conventional usage in classification tasks; here we adopt it in the relative sense commonly used in the 6D pose estimation literature. Specifically, for rotation estimation, we introduce an object viewpoint encoding-based approach that relies solely on synthetic data for training, thereby eliminating the need for large-scale real-world annotations. In translation estimation, our method builds upon GDR-Net [17], employing intermediate geometry features based on dense correspondences for direct regression. This decoupled network design enables the translation component to be trained using real data alone, without the need to learn rotation information, thereby significantly reducing the overall requirement for real-world samples. Experimental results demonstrate that, under few-shot conditions, our method outperforms current state-of-the-art approaches on the LINEMOD [18], LM-O [19], and YCB-V [20] datasets. The implementation is publicly available at <https://github.com/cp-0510/fs6d.git> (accessed on 26 January 2026).

2. Problem Formulation

We study the problem of few-shot 6D object pose estimation from a single RGB image. Given a monocular RGB image containing a target object and its corresponding 3D mesh model, the goal is to estimate the object's 6D pose in the camera coordinate system. Formally, let $I \in \mathbb{R}^{H \times W \times 3}$ denote the input RGB image, and let $\mathcal{M}$ represent the known 3D mesh model of the target object. The 6D pose of the object is defined as $P = (R, t)$, where $R \in SO(3)$ denotes the 3D rotation matrix, and $t \in \mathbb{R}^3$ denotes the 3D translation vector.

3. Related Works

Existing monocular RGB-based 6D pose estimation methods fall into three categories: keypoint-based, template-based, and regression-based approaches.

Keypoint-based methods typically begin by employing convolutional neural networks to detect 2D keypoints of target objects in images and establish 2D–3D correspondences, followed by 6D pose estimation using the PnP algorithm. Early work by Pavlakos et al. [21] introduced semantic keypoint detection to avoid the computational overhead of point-wise matching; however, its performance degrades when dealing with small objects or severe occlusions. To improve robustness under such conditions, PVNet [22] proposes a vector voting mechanism that enables the inference of occluded keypoints, leading to more reliable pose estimation in cluttered scenes. Instead of directly detecting semantic keypoints, BB8 [23] regresses the 2D projections of the eight corners of a 3D bounding box, allowing for efficient pose estimation and improved generalization to unseen object categories. More recently, OK-POSE [24] learns cross-view consistent 3D keypoints from image pairs with relative pose supervision, thereby reducing the dependence on explicit 3D annotations or CAD models while maintaining competitive pose accuracy. Beyond pose accuracy, recent studies have also started to examine the robustness of keypoint-based pose estimation pipelines. In particular, Luo et al. [25] propose a system-level robustness certification

framework for two-stage keypoint–PnP methods, which bounds pose estimation errors by certifying keypoint perturbations under semantic input disturbances.

Template-based methods estimate object poses by pre-constructing 3D models or feature templates and matching observed image features to these templates. For example, LatentFusion [26] significantly improves generalization to unseen objects by incorporating 3D shape learning and multi-view modeling. DPOD [27] combines object detection with dense correspondence matching, enabling pose estimation without requiring precise instance segmentation and demonstrating strong robustness under occlusion and illumination variations, although its performance degrades on textureless objects. PoseRBPF [28] performs pose tracking within a particle filtering framework and is able to maintain stable and continuous pose estimates even in the presence of motion blur and occlusion. RNNPose [29] estimates correspondences between rendered and observed images and integrates a differentiable pose optimization process to iteratively refine object poses, achieving robust and accurate pose refinement even under large initialization errors or severe occlusions. Overall, while template-based methods exhibit advantages in texture-sparse or occluded scenarios, they often incur high computational cost and rely on high-quality template libraries or strong model priors, which limits their practicality in highly cluttered real-world scenes. Inspired by recent advances in causal modeling for image restoration, approaches such as CausalSR [30] demonstrate that incorporating structural causal models and counterfactual reasoning can lead to more robust and interpretable representations, suggesting promising research directions for 6D pose estimation under complex and data-scarce conditions.

Regression-based methods formulate 6D object pose estimation as a direct regression problem, predicting object poses from RGB images without relying on intermediate keypoint representations. Early approaches such as PoseNet [31] adopt end-to-end convolutional neural networks for pose regression but suffer from limited generalization. To improve estimation accuracy, PoseCNN [20] incorporates 2D centroid prediction with a symmetry-aware loss, while still relying on ICP-based refinement, which introduces considerable computational overhead. To improve inference efficiency, Deep-6DPose [32] leverages Mask R-CNN to perform end-to-end pose regression, achieving improved inference speed. Recent research has increasingly shifted toward geometry-guided direct regression to alleviate the implicit and highly non-linear mapping between image features and pose parameters. Geo6D [33] explicitly embeds geometric constraints into the regression process through a residual pose formulation, improving training stability and reducing dependence on large-scale training data. Building upon this idea, GDRNPP [34] proposes a fully learning-based, geometry-guided pose estimation framework with optional differentiable pose refinement, achieving state-of-the-art accuracy and efficiency on the BOP benchmark without relying on traditional optimization pipelines. Despite their efficiency and end-to-end nature, direct regression methods remain sensitive to rotation representation ambiguities and unseen pose distributions.

In recent years, research on 6D object pose estimation has increasingly focused on improving model generalization to unseen objects. These approaches typically target zero-shot pose estimation, aiming to infer object poses without relying on object-specific training, using only image observations combined with external priors. Representative methods include ZeroPose [35], which leverages CAD models as priors during inference for rapid deployment on novel objects; CLIP-6D [36], which exploits vision–language foundation models to learn generalizable object representations and enhance geometric reasoning; and PoseDiffusion [37], which introduces diffusion models within a coarse-to-fine generative framework to progressively estimate and refine poses of unseen objects. Collectively, these works provide effective solutions for 6D pose estimation in scenarios involving previously

unseen objects. Although the above-mentioned methods continue to be optimized and improved at the algorithmic level, their performance gains are severely constrained by the scarcity of real-world annotated data, which serves as the starting point for our research.

4. Methodology

4.1. Overview

In the field of 6D pose estimation, existing methods typically adopt a unified framework to jointly optimize rotation and translation. With sufficient training data, such joint modeling can effectively share feature representations and leverage multi-task information to achieve high performance. However, in few-shot scenarios, the differences between rotation and translation in terms of information sources, learning difficulty, and supervision requirements become pronounced, making joint optimization more sample-intensive and prone to inter-task interference. Specifically, rotation estimation primarily relies on discriminative visual cues provided by object appearance and viewpoint variations, whereas translation estimation heavily depends on real-world geometric structure, object scale, and camera intrinsics, making it more sensitive to high-quality annotated data.

Based on this observation, we propose a decoupled network architecture for few-shot 6D pose estimation, as illustrated in Figure 1. The pipeline begins with an object detector extracting a 128×128 ROI from the RGB image, followed by independent estimation of rotation and translation. Rotation is estimated via a viewpoint encoder trained on synthetic data, while translation leverages geometry-aware features for direct regression, minimizing real data dependency.

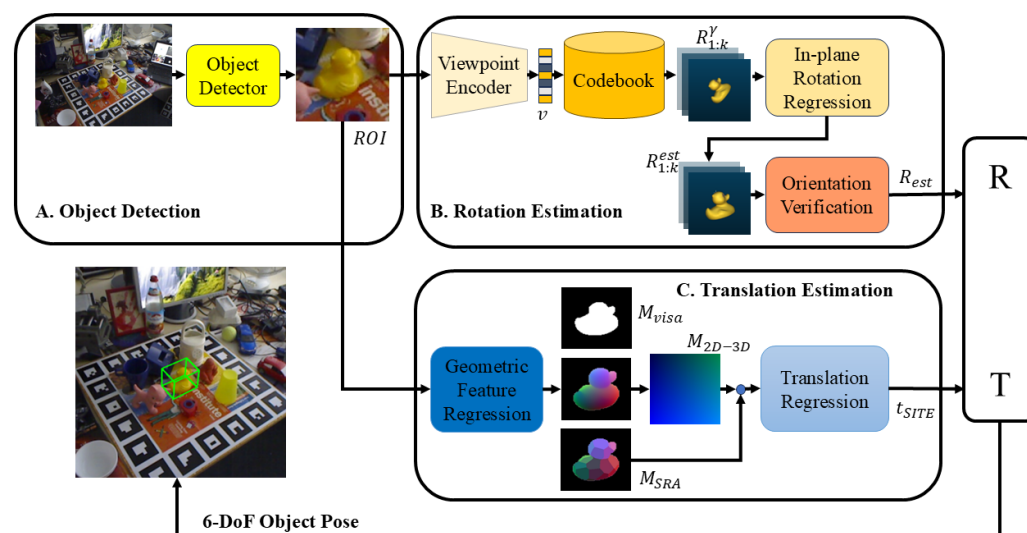


Figure 1. Network Architecture Overview. The pipeline consists of three main components: (A) an object detector that extracts region-of-interest (ROI) patches from input RGB images; (B) a decoupled rotation estimation branch that utilizes a viewpoint encoder trained on synthetic data to retrieve top-K candidate viewpoints from a codebook, followed by in-plane rotation regression and orientation verification to produce the final 3D rotation; and (C) a translation estimation branch that predicts dense geometric features, including 2D–3D correspondences and attention masks, and regresses the translation vector using a lightweight geometry-aware network.

In the rotation estimation network (in Section 4.2), the viewpoint encoder is first employed to construct an object-specific viewpoint codebook, from which multiple candidate viewpoints are retrieved. For each retrieved candidate, an in-plane 2D rotation regression is performed to obtain a complete set of 3D rotation estimates. Finally, a consistency score

is computed for each rotation hypothesis, and the one with the highest score is selected as the final 3D rotation estimate (see Figure 1B).

In the translation estimation network (in Section 4.3), several intermediate geometry features based on dense correspondences are predicted by the network and used to directly regress the full 3D translation (see Figure 1C).

4.2. Viewpoint-Encoded Rotation Matrix Estimation

The rotation estimation network employs a viewpoint encoder to construct an object-specific codebook from synthetic data.

The core of the network is an object viewpoint encoder based on RGB images, which robustly encodes the object viewpoint into a feature vector. In this paper, the object viewpoint is defined as the out-of-plane rotation R^γ , which determines the viewing direction of the object in 3D space. The encoded representation is sensitive to changes in viewpoint while remaining invariant to in-plane rotations around the camera optical axis. Specifically, we adopt a rotation decomposition strategy that factorizes the complete 3D rotation R into the out-of-plane rotation R^γ and the in-plane rotation R^θ , where R^θ represents the residual axial rotation about the camera optical axis:

$$R = R^\theta R^\gamma \quad (1)$$

As shown in Figure 2, this decomposition allows the network to first extract stable viewpoint information using the object viewpoint encoder. It then compensates for axial rotations through an in-plane rotation regression module, thereby achieving complete 3D rotation estimation. This decomposition ensures robustness to in-plane rotations while accurately capturing the full 3D orientation.

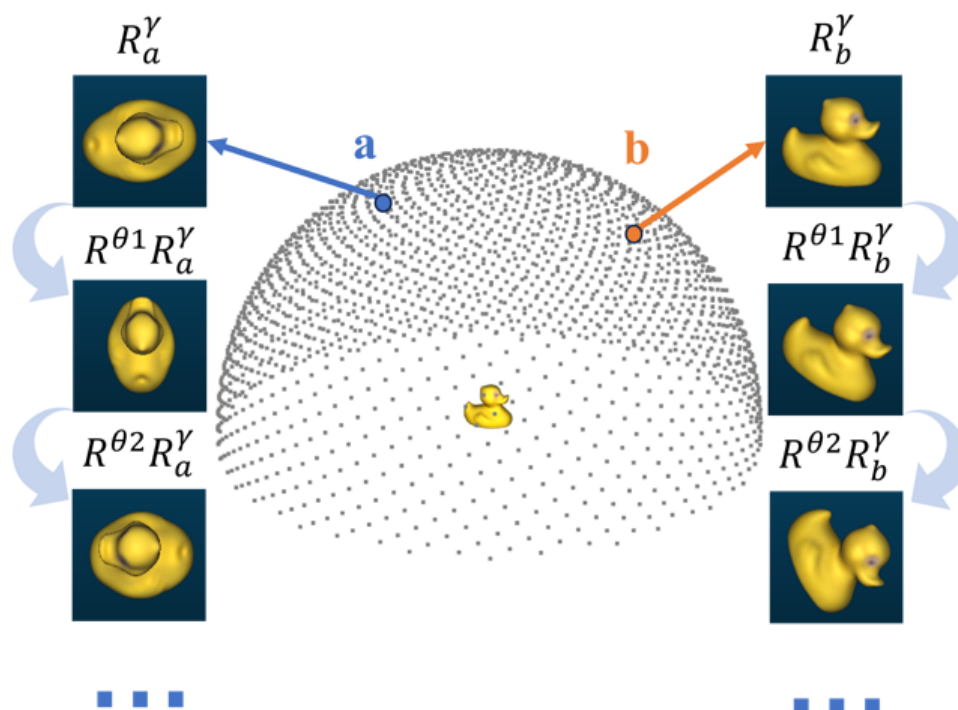


Figure 2. A total of 4000 viewpoints are uniformly sampled on a sphere centered around the object. Two representative viewpoints are selected and denoted as a (R_a^γ) and b (R_b^γ). For each viewpoint, images are synthesized under three different in-plane rotation angles (R^θ) to illustrate the effect of viewpoint and in-plane rotation decomposition.

The entire network consists of three core functional modules: the object viewpoint encoder, in-plane rotation regression, and 3D direction verification. The following sections provide a detailed description of the design and implementation of each of these modules.

4.2.1. Object Viewpoint Encoder

The viewpoint encoder comprises a backbone network and an encoding head H_{vp} . The backbone consists of eight convolutional layers (Conv2D) with batch normalization (BN), while the encoding head is composed of a single convolutional layer (Conv2D), a pooling layer, and a fully connected layer (FC). Using detected ROI images as input, it outputs a 64-dimensional feature vector representing the encoded camera viewpoint.

To achieve invariance to in-plane rotations around the camera's optical axis while maintaining sensitivity to changes in camera viewpoint, we employ a similarity ranking-based loss function. During training, a triplet training sample $\{V, V_\theta, V_\gamma\}$ is constructed using a synthetic RGB dataset rendered from ShapeNet [38], where V is the original RGB image from a canonical viewpoint, V and V_θ differ only in in-plane rotation, and V_γ is obtained from a different camera viewpoint. The corresponding viewpoint feature representations $\{v, v_\theta, v_\gamma\}$ are then extracted using the viewpoint encoder, as shown in Figure 3A. The equivalent loss function L_{vp} is:

$$L_{vp} = \max(\cos(v, v_\gamma) - \cos(v, v_\theta), 0) \quad (2)$$

This encourages the encoder to learn more robust viewpoint representations and ensures good generalization across different objects.

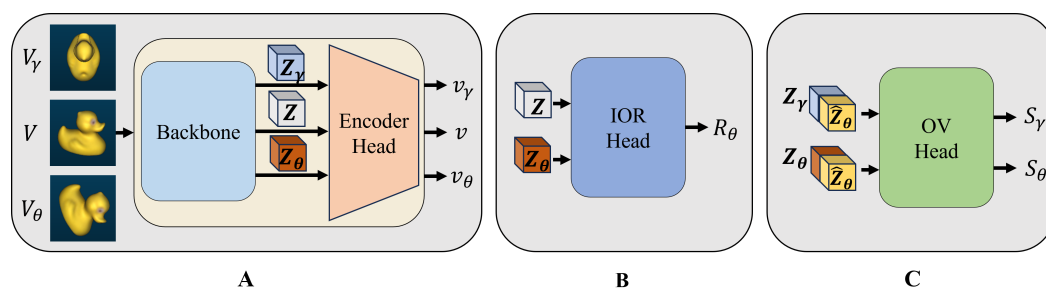


Figure 3. Training strategy of the rotation estimation network, which includes three sub-networks to be trained: (A) Backbone feature extractor with an encoder head; (B) In-plane rotation regression branch; (C) 3D orientation validation branch.

During inference, the trained viewpoint encoder is first used to construct a viewpoint codebook for the target object, which is then utilized for subsequent viewpoint retrieval and in-plane rotation estimation. The specific approach is as follows: a 3D bounding sphere is constructed centered on the target object, with its radius defined as $d = k \times \text{diameter}$, where diameter denotes the geometric diameter of the object's 3D mesh model. In this work, we set $k = 5$, and the detailed justification and experimental analysis are provided in Section 5.4. A total of $N = 4000$ viewpoints $\{R_i\}_{i=1}^N$ are uniformly sampled on the surface of the sphere. Then, each sampled viewpoint is combined with the 3D mesh model of the target object to render a synthetic RGB image set $\{V_i^{syn}\}_{i=1}^N$. Finally, the trained viewpoint encoder is used to extract the corresponding viewpoint feature representations $\{v_i\}_{i=1}^N$ from these images (i.e., first, the backbone network extracts feature maps from the images, then the encoding head encodes them into viewpoint feature vectors). These are stored in a codebook database as a set $\{(v_i, R_i)_{i=1}^N, O_{mesh}, O_{id}\}$, where O_{mesh} is the object mesh model and O_{id} is the object ID, as shown in Figure 4.

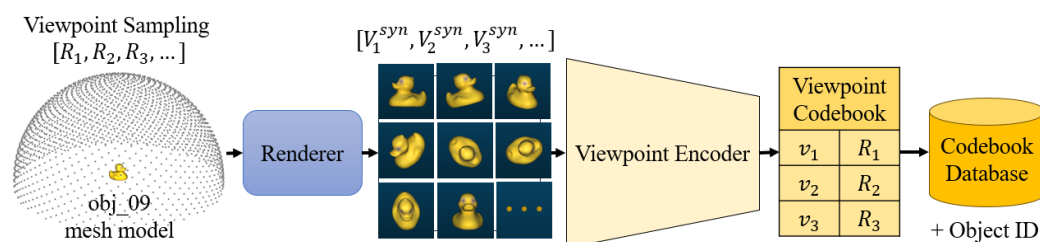


Figure 4. Construction of the object viewpoint codebook.

After constructing the viewpoint codebook for the target object, the viewpoint encoder is used to extract the target viewpoint representation v from the ROI image V . Subsequently, the cosine similarity score between v and all entries in the corresponding viewpoint codebook (indexed by the known object ID) is calculated. The entry with the highest similarity score R_k^γ is then selected as the closest viewpoint. Additionally, the entries can be ranked in descending order based on the cosine similarity scores, and the top K candidate viewpoints $\{R_k^\gamma\}_{k=1}^K$ can be selected from the codebook.

4.2.2. In-Plane Rotation Regression

After obtaining the viewpoint information, the network regresses the in-plane rotation around the camera optical axis, thereby completing the full rotation prediction. To achieve this, we construct the regression network by attaching a regression head H_{IOR} to the backbone network shared with the viewpoint encoder. The regression head consists of a convolutional layer (Conv2D) followed by two fully connected layers (FC). This module takes as input a pair of feature maps corresponding to the same viewpoint but with different in-plane orientations, and regresses the relative in-plane rotation angle.

During inference, we first use the shared backbone network to extract a pair of feature maps $\{z, z_k^{syn}\}$ from the RGB image pair $\{V, V_k^{syn}\}$, where V_k^{syn} is the synthesized RGB image rendered using the retrieved viewpoint R_k^γ . Next, the regression module estimates the in-plane 2D rotation matrix R_k^θ from the feature map pair, and generates the complete 3D orientation estimation via $R_k^{est} = R_k^\theta R_k^\gamma$. Similarly, we can perform in-plane rotation regression for each of the retrieved candidate viewpoints to obtain multiple complete 3D orientation estimations, $\{R_k^{est}\}_{k=1}^K$.

We train this module by minimizing the discrepancy between the ground-truth rotation matrix $\bar{R}_\theta$ and the predicted $\hat{R}_\theta$ (see Figure 3B). Specifically, we use the negative logarithm of cosine similarity to measure the discrepancy and define the loss function L_θ as follows:

$$L_\theta = -\log \left(\frac{(1.0 + \cos(F(T_{\hat{R}_\theta}(V)), F(T_{\bar{R}_\theta}(V))))}{2.0} \right) \quad (3)$$

where F denotes the flattening operation, and T_{R_θ} represents the 2D spatial transformation associated with R_θ [39]. By optimizing this loss, the regression network is encouraged to accurately predict the in-plane rotation and avoid large rotational errors.

4.2.3. 3D Orientation Validation

Based on the previous two modules, multiple complete 3D orientation hypotheses $\{R_k^{est}\}_{k=1}^K$ can be derived. To select the one closest to the ground truth pose, we employ an orientation verification module to estimate the consistency between each candidate object and the actual object orientation in the ROI image V . Using these consistency scores, the multiple hypotheses can be ranked accordingly. Similar to the regression module,

the verification module is constructed by attaching a verification head H_{OV} to the shared backbone. The verification head consists of two convolutional layers (Conv2D), a pooling layer, and a fully connected layer (FC).

During training, we optimize this module using a ranking-based loss function. As shown in Figure 3C, the estimated in-plane rotation R_θ is first applied to the feature map z via a spatial transformation, i.e., $\hat{z}_\theta = T_{R_\theta}(z)$, where T_{R_θ} denotes the 2D spatial transformation corresponding to R_θ [33]. Then, $\hat{z}_\theta$ is concatenated with z_γ and z_θ along the feature channel dimension, respectively, and the concatenated features are fed into H_{OV} to produce consistency scores s_γ and s_θ . The equivalent loss function is:

$$L_{ov} = \max(s_\gamma - s_\theta, 0) \quad (4)$$

This ensures that the correct direction hypothesis ranks higher in the consistency scoring, thereby gradually guiding the model to learn accurate direction verification capability during training.

During inference, the estimated in-plane rotation matrix R_k^θ is first used to transform the feature map z_k^{syn} from the retrieved viewpoint. Next, it is combined with the feature map z from the ROI image and fed into the verification head, enabling the calculation of a consistency score for each 3D orientation hypothesis. Based on the estimated scores, we sort all hypotheses $\{R_k^{est}\}_{k=1}^K$ in descending order and select the top $P \in [1, K]$ orientation hypotheses $\{R_p^{est}\}_{p=1}^P$ as the output. In this work, we set the number of candidate viewpoints to $K = 10$ and use $P = 1$ during inference, retaining only the hypothesis with the highest consistency score as the final pose estimate. All quantitative results reported in this paper are based on this setting.

4.3. Translation Vector Estimation

Translation estimation builds on GDR-Net, predicting intermediate geometric features (e.g., a 64×64 dense correspondence map via ResNet-34) and regressing 3D translation using a lightweight network with three convolutional and fully connected layers. This design ensures stable performance with minimal real data.

The core of this network consists of two modules: the geometric feature regression module and the translation regression module. The following sections will introduce each of them in detail.

Geometric Feature Regression Module: As a prior knowledge extractor for translation regression, this module follows the design of GDR-Net. During inference, features are extracted from the ROI image obtained by detection using a ResNet-34 network. The network then predicts three intermediate geometric feature maps, each of size 64×64 : the dense correspondence map (M_{2D-3D}), the surface region attention map (M_{SRA}), and the visible object mask (M_{visa}), where:

The dense correspondence map (M_{2D-3D}) is obtained by first rendering the 3D object model to estimate the underlying dense coordinate mapping (M_{XYZ}), which is then stacked onto the corresponding 2D pixel coordinates to generate the final map.

The surface region attention map (M_{SRA}) is extracted from M_{XYZ} using farthest point sampling to capture the surface region. These geometric feature maps, by deeply exploring the object's geometric information, provide support for the subsequent 3D translation regression.

Translation Regression Module: It is a lightweight network consisting of three Conv2D layers with group normalization [40] and three fully connected layers (FC). It functions as the core component of the entire sub-network. During the inference phase, the scale-invariant translation parameters $t_{SITE} = [\delta_x, \delta_y, \delta_z]^T$ [41] are directly re-

gressed from M_{2D-3D} and M_{SRA} , ultimately allowing for precise recovery of the object's 3D translation.

4.4. Network Loss Function

The rotation and translation estimation networks are trained end-to-end with independent loss functions to more precisely constrain their respective learning objectives, ensuring that the optimization processes of the two do not interfere with each other.

The rotation estimation network consists of three sub-networks to be trained, and their corresponding optimization methods have been described in Section 4.2. These include the viewpoint encoding loss L_{vp} , the in-plane rotation regression loss L_θ , and the orientation verification loss L_{ov} . Therefore, the overall loss function L_R is defined as:

$$L_R = \frac{1}{N} \sum_{i=1}^N (\lambda_1 L_{vp} + \lambda_2 L_{ov} + \lambda_3 L_\theta) \quad (5)$$

where N denotes the batch size, and λ_1 , λ_2 , and λ_3 are weighting factors. In our experiments, we set the weights to $\lambda_1 = 100$, $\lambda_2 = 10$, $\lambda_3 = 1$; they are empirically chosen via validation.

For optimizing the translation estimation network, we adopt the loss function design of GDR-Net. Specifically, the overall translation loss L_T consists of two components: the geometric consistency loss L_{Geom} and the translation parameter regression loss L_t :

$$L_T = L_{Geom} + L_t \quad (6)$$

The geometric consistency loss L_{Geom} is composed of the L_1 loss computed on the normalized dense correspondence map M_{XYZ} and the visible object mask M_{visa} , along with the cross-entropy (CE) loss applied to the surface region attention map M_{SRA} :

$$L_{Geom} = \|\bar{M}_{vis} \odot (\hat{M}_{XYZ} - \bar{M}_{XYZ})\|_1 + \|\hat{M}_{vis} - \bar{M}_{vis}\|_1 + CE(\bar{M}_{vis} \odot \hat{M}_{SRA}, \bar{M}_{SRA}) \quad (7)$$

On this basis, the scale-invariant 2D object center (δ_x, δ_y) and depth (δ_z) are further supervised individually, forming the translation parameter regression loss L_t :

$$L_t = \|(\hat{\delta}_x - \bar{\delta}_x, \hat{\delta}_y - \bar{\delta}_y)\|_1 + \|\hat{\delta}_z - \bar{\delta}_z\|_1 \quad (8)$$

In the above equations, $\hat{\cdot}$ and $\bar{\cdot}$ denote the predicted and ground truth values, respectively, while $\odot$ indicates element-wise multiplication.

5. Experiment

We evaluate our method on LINEMOD (13 objects in cluttered scenes), LM-O (8 occluded objects), and YCB-V (21 objects with severe occlusions). The ADD(-S) metric, with a 10% diameter threshold, assesses pose accuracy.

5.1. Datasets and Metrics

Dataset: ShapeNet [38] is primarily used to provide training data for rotation estimation. To ensure efficient data loading, we excluded large-scale objects from the dataset, resulting in a reduced set of 19K shapes from the original 52,274 models. For each object, 16 anchor viewpoints are randomly sampled and distributed on a sphere centered at the object. Then, for each anchor viewpoint, a random in-plane rotation and a random out-of-plane rotation are applied. The corresponding RGB images are synthesized using the Pyrender, resulting in a set of 128 viewpoint triplets per object.

LINEMOD [18] is a widely adopted benchmark for single-object 6D pose estimation, comprising RGB-D images and corresponding 3D models for 13 objects captured in cluttered environments. However, as recent works have reported recall rates exceeding 90%, the dataset has become increasingly saturated and less discriminative for evaluating advanced methods. Consequently, we employ the more challenging LM-O [19] and YCBV [20] datasets to better assess the robustness and generalization capability of our approach.

The LM-O dataset consists of 1214 test images and serves exclusively for evaluation purposes. It provides annotated 6D poses for eight objects under partial occlusion, thereby introducing increased challenges for pose estimation. In contrast, the YCB-V dataset comprises a significantly larger set of real-world images containing 21 objects. While YCB-V offers more extensive annotated training data, it poses additional difficulties due to severe object occlusions and the presence of multiple geometrically symmetric objects within cluttered scenes.

During training, approximately 1.4k real images per object from the LM dataset are used as the real training data for LM-O. Although YCB-V offers a larger number of annotated real images, the amount of real data remains limited for most deep learning-based methods. Therefore, a large number of physically-based rendered (PBR) training images [42] are employed to augment the training process and enhance generalization performance. In our experimental setup, the rotation estimation network is trained using solely synthetic images, while the translation estimation requires only 600 real images per object for training, significantly reducing on real-world data. The 600 real images per object used for training are directly sampled from the training splits of public benchmark datasets, without imposing additional prior constraints on background complexity, illumination conditions, or imaging noise. For training a single object, the model contains approximately 4.8 M parameters, corresponding to an apparent training-samples-to-parameters ratio of about 1.25×10^{-4} , i.e., roughly one real training sample per 8000 parameters. It should be noted, however, that the rotation branch is trained solely on synthetic data and kept fixed, while the translation branch is trained using real images. As a result, the number of parameters actually trained on real data is much smaller than the total parameter count.

Metrics: We adopt the widely used ADD(-S) metric to evaluate our method. This metric measures the average distance between model points projected using the predicted pose and those projected using the ground-truth pose in the camera coordinate frame. In all experiments presented in this paper, a predicted pose is considered correct if the ADD(-S) error is less than 10% of the object's diameter, which is the most commonly used threshold. For the YCB-V dataset, we additionally report the Area Under the Curve (AUC) of the ADD(-S) metric, with a maximum threshold of 10 cm.

Network Training: The experiments were conducted using the PyTorch deep learning framework (version 1.10). The Adam optimizer was employed for weight updates, with a cosine annealing learning rate schedule ranging from 1×10^{-3} to 1×10^{-5} and a weight decay of 1×10^{-5} . Training was performed on a single Nvidia RTX 4060 GPU, and the overall training process required approximately 4 days. During inference, the model requires approximately 4.3 GB of system memory and takes on average 1.03 s to process each image.

5.2. Experiments on LINEMOD

Ablation Study on the Number of Viewpoints: To systematically evaluate the impact of the number of viewpoints on rotation retrieval accuracy and overall pose estimation performance, we conduct an ablation study on the LINEMOD (LM) dataset. In this experiment, the rotation estimation module directly takes the predictions of the object detector as input, while all other components in the rotation estimation pipeline are kept unchanged,

including the pre-trained viewpoint encoder, the in-plane rotation regression module, and the 3D verification module. This design ensures that any performance variation can be solely attributed to differences in the number of viewpoints N .

Table 1 presents the ablation study results on the LINEMOD (LM) dataset with respect to the number of viewpoints N . The viewpoint retrieval accuracy is measured as the percentage of predictions with a rotation error smaller than 10° . The codebook construction is performed once in the offline stage and is therefore excluded from the per-frame inference cost. The table also reports the codebook construction time (in seconds), the ADD(-S) scores under different values of N , and the per-frame retrieval time (ms/frame).

Table 1. Ablation study on the LINEMOD (LM) dataset with respect to the number of viewpoints N .

N	Retrieval Accuracy	ADD(-S)	Construction Time	Retrieval Time
500	42.7	64.3	12.5	7
1000	51.6	71.4	28.3	19
2000	63.6	85.4	47.9	44
4000	72.4	94.8	104.2	92
8000	75.7	95.0	246.3	187

As shown in the table results, when the viewpoint sampling is sparse (e.g., $N = 500$ and $N = 1000$), the rotation space is insufficiently covered and the angular gaps between neighboring viewpoints are large, resulting in low viewpoint retrieval accuracies of only 42.7% and 51.6%, respectively, with the corresponding ADD(-S) scores being clearly limited. As the number of viewpoints increases to $N = 2000$, the rotation space is discretized more densely, leading to significant improvements in both viewpoint retrieval accuracy and ADD(-S). This indicates that denser viewpoint sampling effectively alleviates the viewpoint missing problem and provides more reliable initialization for subsequent rotation refinement and geometric verification. Further increasing N to 4000 continues to improve performance, with ADD(-S) reaching 94.8%; however, the performance gain becomes marginal compared to the previous stage. When N is further increased to 8000, the performance improvement saturates, while the per-frame retrieval time grows almost linearly, revealing a clear diminishing-returns effect.

Overall, overly coarse viewpoint sampling severely degrades rotation retrieval and pose estimation accuracy, whereas excessively dense sampling introduces unnecessary computational overhead. Therefore, a trade-off between rotation resolution and efficiency is required. Based on the above experimental results, we finally select $N = 4000$ as the default number of viewpoints, achieving a favorable balance between accuracy and computational cost.

Evaluation of the Core Modules in the Rotation Estimation: This study focuses on evaluating the rotation estimation network by independently assessing its core sub-modules. Experiments on the LINEMOD (LM) dataset examine each module's accuracy and robustness under different input conditions. Specifically, we compare performance using predicted ROIs (Pred-ROI) from object detectors and ground-truth ROIs (GT-ROI) from annotations to understand their influence on overall system effectiveness.

Figure 5 illustrates the performance of the viewpoint retrieval module on the LM dataset across multiple error thresholds, comparing results based on predicted ROIs and ground-truth ROIs. In this evaluation, only the top-scoring pose hypothesis is considered to ensure a fair and focused assessment of retrieval accuracy. The results show that 72% of cases achieve an angular error $\leq 10^\circ$, confirming the high reliability of the viewpoint retrieval module. While performance with predicted ROIs is slightly lower than with ground-truth ROIs, the gap is small, suggesting limited sensitivity to ROI quality. Addition-

ally, results on synthetic and real images are comparable, indicating strong cross-domain generalization and robustness to domain shifts.

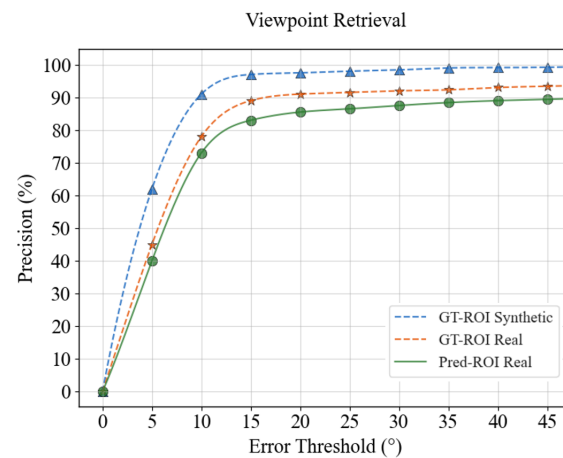


Figure 5. Evaluation of the viewpoint retrieval module. “Synthetic” refers to ROIs extracted from synthetic RGB images, while “Real” indicates ROIs obtained from real-world RGB images. “GT” and “Pred” represent ground-truth and predicted results, respectively.

Building on the results of viewpoint retrieval, we proceed to evaluate the impact of viewpoint accuracy on the in-plane rotation estimation module. Specifically, we compare two conditions: using predicted viewpoints retrieved from the codebook and using ground-truth viewpoints derived from annotations. As shown in Figure 6, the proposed in-plane rotation regressor achieves 74% accuracy within a $\leq 10^\circ$ error margin when using predicted viewpoints and ROIs, and surpasses 90% accuracy when ground-truth viewpoints are provided. These results confirm that in-plane rotation estimation is highly dependent on the quality of viewpoint retrieval, underscoring its critical role within the decoupled rotation estimation pipeline.

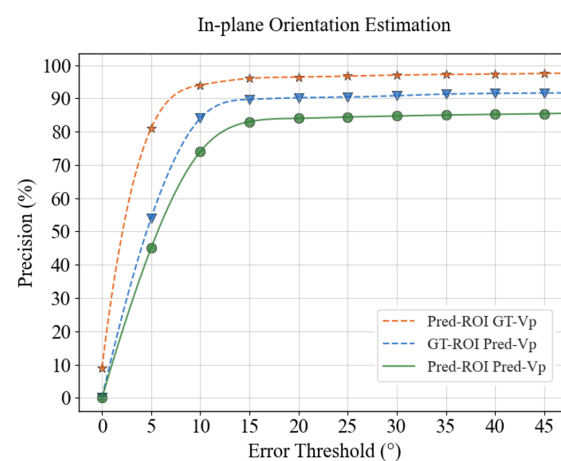


Figure 6. Evaluation of the in-plane rotation estimation module. “GT-Vp” and “Pred-Vp” represent the ground truth viewpoint based on the actual pose and the viewpoint predicted by the viewpoint retrieval module, respectively.

Overall evaluation of the proposed method: To further validate the effectiveness of the proposed method in practical 6D pose estimation tasks, we conducted a systematic evaluation of the complete model on the LINEMOD (LM) dataset.

Table 2 shows our method achieving 94.8% ADD(-S) with Fast R-CNN ROIs and 97.6% with ground-truth ROIs, highlighting robustness to detector errors.

Table 2. Experimental results of the proposed method on the LM, where (-) indicates that the feature was not used.

Method	Detector (Test)	ADD(-S)
Ours	Fast R-CNN	94.8
Ours (GT)	-	97.6

From the results in the table, it can be observed that when using Fast R-CNN as the object detector to provide the ROI (Pred-ROI) for testing, the model achieved an ADD(-S) accuracy of 94.8%. This demonstrates that, when combined with existing detectors, the proposed pose estimation method exhibits strong practicality and robustness. Furthermore, when the input is the ground truth annotated ROI (GT-ROI) provided by the dataset, the accuracy improves to 97.6%, reflecting the performance upper limit of the proposed network under high-quality input conditions.

It is worth noting that, although the introduction of the object detector introduces some errors, the overall performance of the model only shows a slight decrease. This indicates that the sub-modules, particularly the rotation estimation component, possess strong adaptability and stability under varying input quality. This result also suggests that the proposed method has good potential for real-world applications.

5.3. Comparison Experiments on LM-O and YCB-V

On LM-O, our method achieves 65.3% ADD(-S), surpassing GDR-Net (62.2%) due to its decoupled design mitigating rotation errors under occlusion. On YCB-V, it excels with 65.9% ADD(-S) and 92.4% AUC of ADD-S, outperforming baselines in multi-object scenarios. Some qualitative results are shown in Figure 7.

In the experiments, the proposed method is systematically compared with several state-of-the-art approaches, primarily those based on monocular RGB data. To ensure a fair comparison, we used Faster R-CNN [43] to extract 2D features from RGB images for ROI extraction in the LM-O dataset tests. For the YCB-V dataset, we used the publicly available detection results provided by FCOS [44] as input to adapt to the detector configuration of this dataset.

Table 3 presents the quantitative comparison results between the proposed method and current state-of-the-art approaches on the LM-O dataset, with ADD(-S) used as the unified performance metric. The best results of GDR-Net are also included, providing a strong reference benchmark for performance comparison. The proposed method can be extended to other unseen instances with available CAD models, and does not address category-level generalization.

Table 3. Comparison with state-of-the-art methods on the LM-O dataset, where (*) indicates symmetric objects.

Method	PVNet [22]	HybridPose [45]	RePose [46]	GDR-Net [17]	Ours
Ape	15.8	20.9	31.1	46.8	50.7
Can	63.3	75.3	80.0	90.8	89.5
Cat	16.7	24.9	25.6	40.5	41.3
Driller	65.7	70.2	73.1	82.6	83.7
Duck	25.2	27.9	43.0	46.9	51.2
Eggbox *	50.2	52.4	51.7	54.2	53.6
Glue *	49.6	53.8	54.3	75.8	78.6
Holepuncher	36.1	54.2	53.6	60.1	74.5
Avg.	40.8	47.5	51.6	62.2	65.3

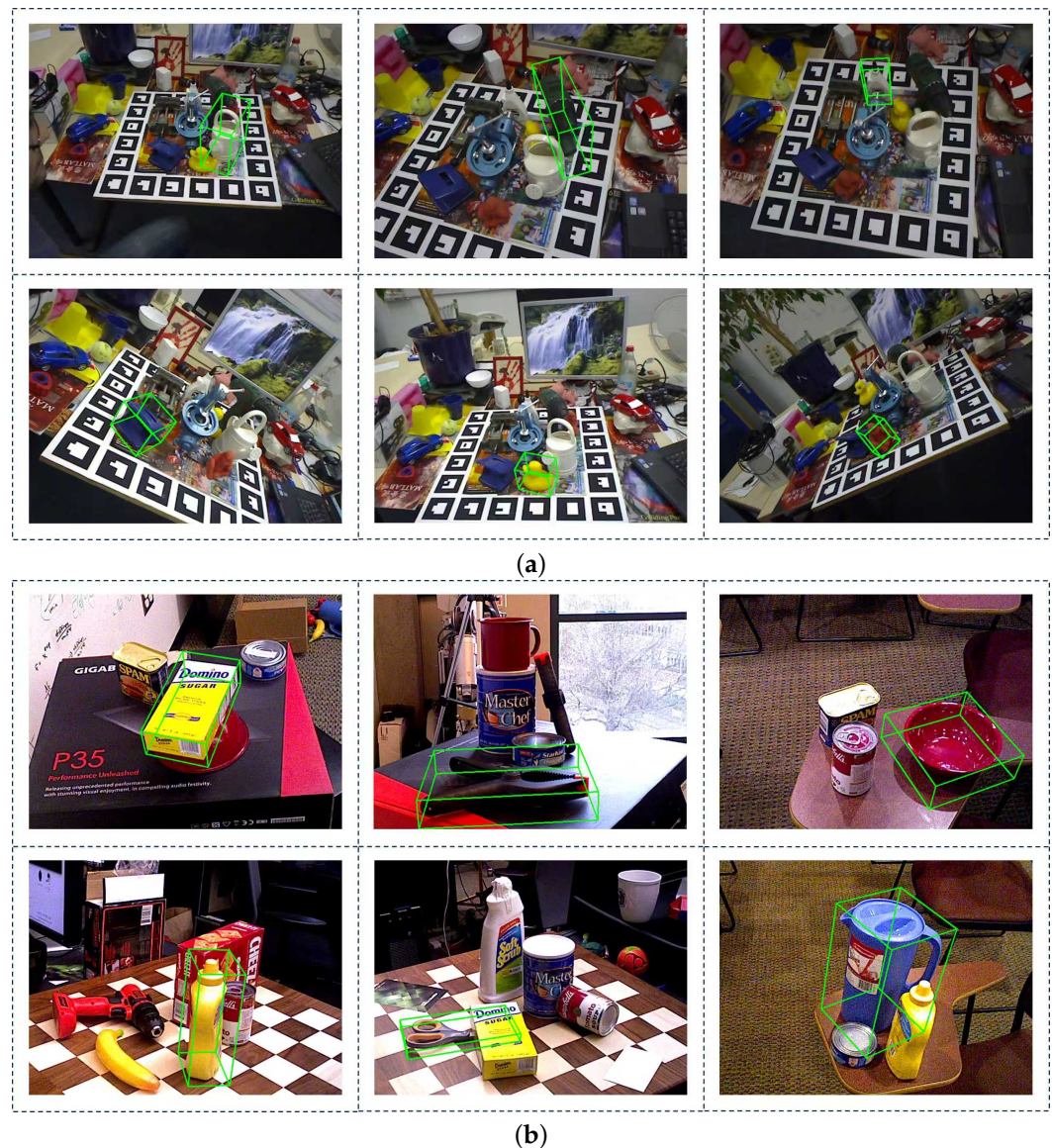


Figure 7. Qualitative results on the LM-O (a) and the YCB-Video (b) datasets.

The results in the table indicate that the proposed method outperforms other approaches across most object categories. In particular, it demonstrates superior robustness and estimation accuracy on objects with complex shapes or significant occlusion, such as Ape, Duck, and Holepuncher. Moreover, for symmetric objects such as Eggbox and Glue, although traditional methods have certain advantages in handling symmetry, the proposed method still achieves further performance improvements on Glue, demonstrating its strong adaptability to symmetry-related challenges. Overall, the proposed method achieved an average accuracy of 65.3% on the LM-O dataset, representing an improvement of 3.1 percentage points over GDR-Net. This clearly demonstrates the stability and robustness of the proposed approach under complex occlusion and challenging interference conditions.

To further evaluate the few-shot capability of our method, we compare it with several recently proposed state-of-the-art approaches. While these methods generally achieve higher accuracy when trained with large-scale real data, for a fair comparison under the same limited supervision setting, all baseline models are retrained (or reported) using 600 real images, consistent with our training setup. We use the official implementations and follow the training protocols recommended by the authors as closely as possible, only

reducing the amount of real training data to ensure a fair comparison under the same supervision budget.

The quantitative results are shown in Table 4. Under this constrained training regime, all baseline methods experience significant performance drops, with average accuracies of 35.9%, 38.2%, 40.5%, 44.7%, and 48.3%, respectively. In contrast, our method maintains consistently strong performance across all object categories, achieving an average accuracy of 65.3% on the LM-O dataset. This highlights the superior generalization ability of our proposed design, particularly in low-data scenarios with occlusion and symmetry challenges.

Table 4. Comparison on LM-O with 600 Real Images, where (*) indicates symmetric objects.

Method	Gen6D [47]	Zebrapose [48]	Checkerpose [49]	GDRNPP [34]	FoundationPose [50]	Ours
Ape	35.2	32.7	35.0	37.2	41.3	50.7
Can	38.4	50.7	55.1	55.3	61.2	89.5
Cat	28.1	26.9	28.3	32.6	34.5	41.3
Driller	37.3	46.1	47.2	49.7	54.2	83.7
Duck	33.2	31.3	34.6	36.5	39.3	51.2
Eggbox *	36.4	39.5	41.5	42.7	46.2	53.6
Glue *	39.1	40.3	42.9	49.8	54.4	78.6
Holepuncher	39.4	38.0	39.3	53.4	55.5	74.5
Avg.	35.9	38.2	40.5	44.7	48.3	65.3

Table 5 presents the quantitative comparison results between the proposed method and current state-of-the-art approaches on the YCB-V dataset. The AUC values reported in the table are computed using full-point interpolation.

Table 5. Comparison with state-of-the-art methods on the YCB-V dataset, where (-) indicates results not reported in the original paper.

Method	ADD(-S)	AUC of ADD-S	AUC of ADD(-S)
PVNet [22]	-	-	73.4
CosyPose [16]	-	89.8	84.5
RePose [46]	62.1	88.5	82.0
GDR-Net [17]	60.1	91.6	84.4
Ours	65.9	92.4	86.9

The results in the table show that PVNet and CosyPose did not report complete ADD(-S) or AUC of ADD-S results in their original papers, which limits the comprehensiveness of the comparison. In contrast, RePose and GDR-Net serve as strong baseline methods with competitive performance. The proposed method demonstrates superior performance across all metrics: achieving 65.9% on ADD(-S), 92.4% on AUC of ADD-S, and 86.9% on AUC of ADD(-S), which represent improvements of 5.8%, 0.8%, and 2.5% over GDR-Net, respectively. In summary, the proposed method demonstrates superior pose estimation performance on the YCB-V dataset, maintaining high accuracy in typical multi-object and occlusion scenarios. It also exhibits strong generalization ability and practical applicability.

Similarly, we extend the comparison on the YCB-V dataset under the few-shot setting, where all baseline methods are retrained (or reported) using only 600 real images, consistent with our training setup. The results are summarized in Table 6. Under this limited supervision, all baseline methods suffer noticeable performance drops, with ADD(-S) scores ranging from 33.7% to 43.5%. In contrast, our method achieves 65.9% ADD(-S) and attains the highest scores in both AUC of ADD-S (92.4%) and AUC of ADD(-S) (86.9%). These

results further demonstrate the robustness and effectiveness of our approach in real-world scenarios where collecting large-scale annotated real data is costly or impractical.

Table 6. Comparison on YCB-V with 600 Real Images.

Method	ADD(-S)	AUC of ADD-S	AUC of ADD(-S)
Gen6D [47]	33.7	60.1	53.9
ZebraPose [48]	35.7	64.7	57.4
CheckerPose [49]	37.8	67.5	58.9
GDRNPP [34]	40.3	69.4	60.8
FoundationPose [50]	43.5	75.4	63.4
Ours	65.9	92.4	86.9

We further report the AUC of ADD(-S) under a much stricter maximum threshold of 1 cm to evaluate high-precision pose estimation. Compared with the commonly used 10 cm threshold, this metric significantly reduces absolute scores for all methods. However, as shown in Table 7, our method consistently maintains a clear margin over all baselines, indicating that the proposed approach is not merely benefiting from a loose tolerance, but is capable of producing more accurate and stable poses under stringent conditions.

Table 7. Comparison under Strict ADD(-S)@1cm Metric on YCB-V with 600 Real Images.

Method	AUC of ADD-S @1cm	AUC of ADD(-S) @1cm
Gen6D [47]	12.4	9.7
ZebraPose [48]	14.6	11.5
CheckerPose [49]	17.2	12.1
GDRNPP [34]	18.4	14.3
FoundationPose [50]	21.5	16.4
Ours	33.8	28.2

5.4. Additional Experiments

Edge Deployment Feasibility and Memory Analysis: To assess the feasibility of deploying the proposed framework on resource-constrained edge devices, we conducted a detailed memory profiling of its main components during inference on an RTX 4060 GPU. Table 8 summarizes the memory footprint of the rotation branch, translation branch, viewpoint codebook, and intermediate feature maps. The analysis shows that the viewpoint codebook and dense geometric feature extraction dominate memory usage, together accounting for over 63% of the total 4.3 GB consumption.

Table 8. Memory usage of key modules during inference.

Module	Memory Usage (MB)
Rotation Branch	927
Translation Branch	701
Viewpoint Codebook	2137
Feature Maps	630
Total	4395

This highlights the primary bottlenecks for edge deployment. While the current un-optimized PyTorch implementation is not suitable for direct use on typical edge chips, the results provide valuable guidance for targeted optimization. Potential strategies include backbone pruning, codebook compression, mixed-precision inference, and offline viewpoint precomputation, all of which could substantially reduce memory and power

requirements. Although direct power measurement on edge hardware remains future work, inference latency and memory consumption serve as practical proxy indicators for expected energy usage. These insights lay a foundation for the development of an edge-optimized variant of the proposed framework.

Impact of Distance Factor on Feature Matching Consistency Across Viewpoints:

This study systematically analyzes the effect of the distance factor k on feature matching accuracy and consistency under different viewpoint conditions. For each value of k , images are rendered at the corresponding distances from fixed viewpoints, and feature vectors are extracted accordingly. The extracted features are then matched against the reference feature vectors of the same viewpoints using cosine similarity, enabling a quantitative evaluation of how variations in the distance factor influence feature representation consistency and matching performance.

The experiments are conducted on a viewpoint feature matching benchmark, where the reference feature vector of each viewpoint serves as the matching standard to ensure consistency and reproducibility. For each value of k , multiple samples are evaluated and the results are averaged to improve the reliability of the experimental conclusions.

The experimental results show that when $k = 5$, the cosine similarity of feature matching reaches its maximum value of 0.99, indicating that this distance factor achieves optimal matching performance while maintaining high feature representation consistency (see Figure 8). In contrast, when k is either too small or too large, the cosine similarity decreases significantly, exhibiting a clear unimodal trend.

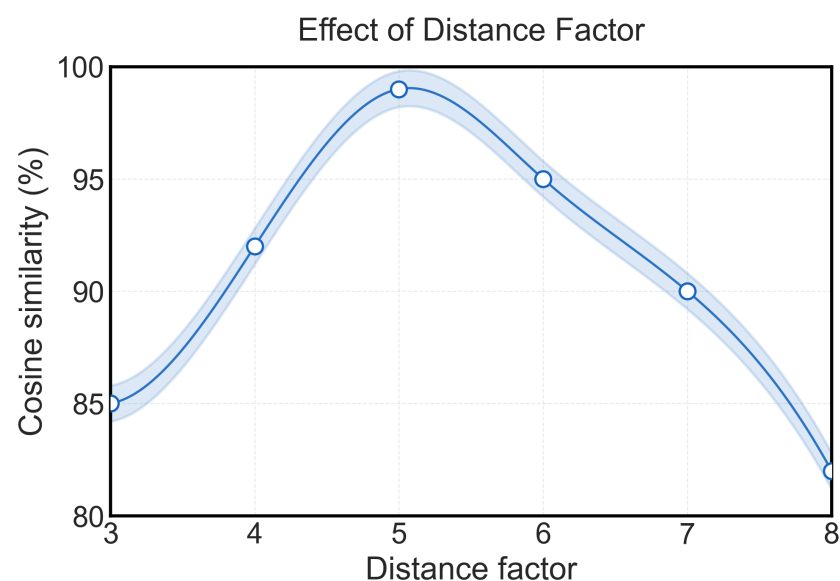


Figure 8. Effect of the distance factor on viewpoint feature matching.

In summary, this experiment verifies the critical role of distance factor selection in viewpoint feature matching and provides quantitative justification for the parameter settings adopted in the subsequent experiments of this work.

Model Capacity and Overfitting Analysis: To further assess the potential risk of overfitting under limited real training data and to analyze the contribution of different components in our framework, we conduct a complementary experiment by constructing a lightweight variant of the proposed model. Specifically, we reduce the channel width of the convolution layers and remove redundant fully connected layers in the translation estimation branch, resulting in an approximately 10× reduction in parameters (from 1.5 M to 0.15 M), while keeping the rotation branch unchanged.

We evaluate both models on the LM dataset using the ADD(-S) metric. As shown in Table 9, the lightweight variant achieves an accuracy of 88.6%, compared to 94.8% for the original model. Although the absolute performance decreases, the degradation remains relatively limited (about 6.2%) despite a 90% reduction in parameters, and the overall performance trend is preserved. These results suggest that the proposed framework does not critically rely on excessive model capacity, and that its robustness primarily stems from the decoupled design and geometry-aware formulation rather than from simply increasing the number of parameters.

Table 9. Evaluation on the LM dataset for the original and lightweight models.

Model	Translation Branch Params	Rotation Branch Params	ADD(-S)
Original	1.5 M	3.3 M	94.8
Lightweight	0.15 M	3.3 M	88.6

6. Conclusions, Limitations and Future Work

This paper proposes a decoupled 6D pose estimation framework for few-shot learning, achieving state-of-the-art performance on the LINEMOD, LM-O, and YCB-V datasets, with only 600 real images required per object.

1. Limitations: This method relies on accurate CAD models, which, to some extent, limits its applicability. The proposed framework is mainly suitable for common scenarios with moderate complexity. Under conditions of heavy occlusion, significant scale ambiguity, or weak texture, the performance of the decoupled rotation and translation estimation strategy may degrade noticeably. In addition, the overall accuracy of the framework partially depends on the performance of the object detector and exhibits limited generalization to unseen object categories.

2. Future Work: Research could explore integrating more robust detection methods, developing end-to-end optimization strategies, and reducing dependency on CAD models, thereby enhancing the framework's robustness in complex scenarios and its generalization to novel objects.

Author Contributions: W.P.: Conceptualization, Methodology, Investigation, Writing—original draft, Writing—review and editing, Project administration, Funding acquisition. P.C.: Methodology, Software, Validation, Investigation, Writing—original draft. L.L.: Resources, Supervision, Project administration, Funding acquisition. Z.S.: Data analysis, Figures. H.Z.: Data collection, Figures. W.Z.: Data analysis, Data interpretation, Figures. G.G.: Literature search, Data collection. P.L.: Data interpretation, Literature search. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (62375078), the Key Research Project Plan for Higher Education Institutions in Henan Province (24ZX011), the Cultivation Programme for Young Backbone Teachers in Henan University of Technology, and the Science and Technology Innovation Project of Chinese Academy of Traditional Chinese Medicine (ZN2024A02).

Data Availability Statement: The data presented in this study are openly available in the BOP Benchmark at <https://bop.felk.cvut.cz/datasets/> (accessed on 26 January 2026).

Conflicts of Interest: Author W.P. was employed by the company OPT Machine Vision Tech Co., Ltd. Author G.G. was employed by the company Mech-Mind Robotics Technologies Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

- Collet, A.; Martinez, M.; Srinivasa, S.S. The moped framework: Object recognition and pose estimation for manipulation. *Int. J. Robot. Res.* **2011**, *30*, 1284–1306. [[CrossRef](#)]
- Zhu, M.; Derpanis, K.G.; Yang, Y.; Brahmbhatt, S.; Zhang, M.; Phillips, C.; Lecce, M.; Daniilidis, K. Single image 3d object detection and pose estimation for grasping. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 3936–3943.
- Tremblay, J.; To, T.; Sundaralingam, B.; Xiang, Y.; Fox, D.; Birchfield, S. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv* **2018**, arXiv:1809.10790. [[CrossRef](#)]
- Manhardt, F.; Kehl, W.; Gaidon, A. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019, Long Beach, CA, USA, 15–20 June 2019; pp. 2069–2078.
- Wu, D.; Zhuang, Z.; Xiang, C.; Zou, W.; Li, X. 6d-vnet: End to-end 6-dof vehicle pose estimation from monocular rgb images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops 2019, Long Beach, CA, USA, 15–20 June 2019.
- Marchand, E.; Uchiyama, H.; Spindler, F. Pose estimation for augmented reality: A hands-on survey. *IEEE Trans. Vis. Comput. Graph.* **2015**, *22*, 2633–2651. [[CrossRef](#)] [[PubMed](#)]
- Tang, F.; Wu, Y.; Hou, X.; Ling, H. 3d mapping and 6d pose computation for real time augmented reality on cylindrical objects. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 2887–2899. [[CrossRef](#)]
- Hodaň, T.; Sundermeyer, M.; Drost, B.; Labbé, Y.; Brachmann, E.; Michel, F.; Rother, C.; Matas, J. Bop challenge 2020 on 6d object localization. In Proceedings of the European Conference on Computer Vision 2020, Glasgow, UK, 23–28 August 2020; pp. 577–594.
- Sun, J.; Wang, Z.; Zhang, S.; He, X.; Zhao, H.; Zhang, G.; Zhou, X. Onepose: One-shot object pose estimation without cad models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2022, New Orleans, LA, USA, 18–24 June 2022; pp. 6825–6834.
- Hodan, T.; Barath, D.; Matas, J. Epos: Estimating 6d pose of objects with symmetries. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, Seattle, WA, USA, 14–19 June 2020; pp. 11703–11712.
- Park, K.; Patten, T.; Vincze, M. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision 2019, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7668–7677.
- Xu, Y.; Lin, K.Y.; Zhang, G.; Wang, X.; Li, H. Rnnpose: Re current 6-dof object pose refinement with robust correspondence field estimation and pose optimization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2022, New Orleans, LA, USA, 18–24 June 2022; pp. 14880–14890.
- Kehl, W.; Manhardt, F.; Tombari, F.; Ilic, S.; Navab, N. Ssd-6d: Makingrgb-based3ddetectionand6dposeestimationgreat again. In Proceedings of the IEEE International Conference on Computer Vision 2017, Venice, Italy, 22–29 October 2017; pp. 1521–1529.
- Shugurov, I.; Li, F.; Busam, B.; Ilic, S. Osop: A multi-stage one shot object pose estimation framework. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2022, New Orleans, LA, USA, 18–24 June 2022; pp. 6835–6844.
- Li, Y.; Wang, G.; Ji, X.; Xiang, Y.; Fox, D. Deepim: Deep iterative matching for 6d pose estimation. In Proceedings of the European Conference on Computer Vision (ECCV) 2018, Munich, Germany, 8–14 September 2018; pp. 683–698.
- Labbe, Y.; Carpentier, J.; Aubry, M.; Sivic, J. Cosypose: Consistent multi-view multi-object 6d pose estimation. In Proceedings of the European Conference on Computer Vision 2020, Glasgow, UK, 23–28 August 2020; pp. 574–591.
- Wang, G.; Manhardt, F.; Tombari, F.; Ji, X. 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 2021, Nashville, TN, USA, 19–25 June 2021; pp. 16611–16621.
- Hinterstoisser, S.; Lepetit, V.; Ilic, S.; Holzer, S.; Bradski, G.; Konolige, K.; Navab, N. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Proceedings of Computer Vision—ACCV 2012*; Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 548–562.
- Brachmann, E.; Michel, F.; Krull, A.; Yang, M.Y.; Gumhold, S.; Rother, C. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 3364–3372.
- Xiang, Y.; Schmidt, T.; Narayanan, V.; Fox, D. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv* **2017**, arXiv:1711.00199.
- Pavlakos, G.; Zhou, X.; Chan, A.; Derpanis, K.G.; Daniilidis, K. 6-dof object pose from semantic keypoints. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2011–2018.
- Peng, S.; Liu, Y.; Huang, Q.; Zhou, X.; Bao, H. Pvnet: Pixel wise voting network for 6dof pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019, Long Beach, CA, USA, 15–20 June 2019; pp. 4561–4570.

23. Rad, M.; Lepetit, V. 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, Venice, Italy, 22–29 October 2017; pp. 3828–3836.
24. Zhang, S.; Zhao, W.; Guan, Z.; Zhao, W.; Peng, J.; Fan, J. Learning cross-view consistent 3D keypoints for object 6D pose estimation. *IEEE Trans. Circuits Syst. Video Technol.* **2025**, *35*, 6816–6831. [[CrossRef](#)]
25. Luo, X.; Wei, T.; Liu, S.; Wang, Z.; Mattei-Mendez, L.; Loper, T.; Liu, C. Certifying robustness of learning-based keypoint detection and pose estimation methods. *ACM Trans. Cyber-Phys. Syst.* **2025**, *9*, 1–26. [[CrossRef](#)]
26. Park, K.; Mousavian, A.; Xiang, Y.; Fox, D. Latentfusion: End to-end differentiable reconstruction and rendering for unseen object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, Seattle, WA, USA, 14–19 June 2020; pp. 10710–10719.
27. Zakharov, S.; Shugurov, I.; Ilic, S. Dpod: 6d pose object detector and refiner. In Proceedings of the IEEE/CVF International Conference on Computer Vision 2019, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1941–1950.
28. Deng, X.; Mousavian, A.; Xiang, Y.; Xia, F.; Bretl, T.; Fox, D. Poserbpf: Arao-blackwellized particle filter for 6-d object pose tracking. *IEEE Trans. Robot.* **2021**, *37*, 1328–1342. [[CrossRef](#)]
29. Xu, Y.; Lin, K.Y.; Zhang, G.; Wang, X.; Li, H. RNNPose: 6-DoF object pose estimation via recurrent correspondence field estimation and pose optimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 4669–4683. [[CrossRef](#)] [[PubMed](#)]
30. Lu, Z.; Lu, B.; Wang, F. CausalSR: Structural causal model-driven super-resolution with counterfactual inference. *Neurocomputing* **2025**, *646*, 130375. [[CrossRef](#)]
31. Kendall, A.; Grimes, M.; Cipolla, R. Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the IEEE International Conference on Computer Vision 2015, Santiago, Chile, 7–13 December 2015; pp. 2938–2946.
32. Do, T.T.; Cai, M.; Pham, T.; Reid, I. Deep-6dpose: Recovering 6d object pose from a single rgb image. *arXiv* **2018**, arXiv:1802.10367. [[CrossRef](#)]
33. Chen, J.; Sun, M.; Zheng, Y.; Bao, T.; He, Z.; Li, D.; Jiang, X. Geo6D: Geometric-constraints-guided direct object 6D pose estimation network. *IEEE Trans. Multimed.* **2025**, *27*, 5770–5783. [[CrossRef](#)]
34. Liu, X.; Zhang, R.; Zhang, C.; Wang, G.; Tang, J.; Li, Z.; Ji, X. GDRNPP: A geometry-guided and fully learning-based object pose estimator. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 5742–5759. [[CrossRef](#)] [[PubMed](#)]
35. Chen, J.; Zhou, Z.; Sun, M.; Zhao, R.; Wu, L.; Bao, T.; He, Z. ZeroPose: CAD-prompted zero-shot object 6D pose estimation in cluttered scenes. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *35*, 1251–1264. [[CrossRef](#)]
36. Wang, H.; Liu, H.; Ren, J.; Tan, M.; Jiang, Z. CLIP-6D: Empowering CLIP as a zero-shot 6D pose estimator through generalizable object-specific representations. In Proceedings of the 33rd ACM International Conference on Multimedia, Dublin, Ireland, 27–31 October 2025; pp. 2566–2575.
37. Zhou, J.; Zhu, Q.; Wang, Y.; Feng, M.; Wu, C.; Liu, X.; Mian, A. PoseDiffusion: A coarse-to-fine framework for unseen object 6-DoF pose estimation. *IEEE Trans. Ind. Inform.* **2024**, *20*, 11127–11138. [[CrossRef](#)]
38. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. Shapenet: An information-rich 3d model repository. *arXiv* **2015**, arXiv:1512.03012.
39. Jaderberg, M.; Simonyan, K.; Zisserman, A. Spatial transformer networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 2017–2025.
40. Wu, Y.; He, K. Group normalization. In Proceedings of the European Conference on Computer Vision (ECCV) 2018, Munich, Germany, 8–14 September 2018; pp. 3–19.
41. Li, Z.; Wang, G.; Ji, X. Cdpm: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision 2019, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7678–7687.
42. Denninger, M.; Sundermeyer, M.; Winkelbauer, D.; Zidan, Y.; Olefir, D.; Elbadrawy, M.; Lodhi, A.; Katam, H. Blenderproc. *arXiv* **2019**, arXiv:1911.01911. [[CrossRef](#)]
43. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2017**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
44. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision 2019, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
45. Song, C.; Song, J.; Huang, Q. Hybridpose: 6d object pose estimation under hybrid representations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, Seattle, WA, USA, 14–19 June 2020; pp. 431–440.
46. Iwase, S.; Liu, X.; Khiroudkar, R.; Yokota, R.; Kitani, K.M. Repose: Fast 6d object pose refinement via deep texture rendering. In Proceedings of the IEEE/CVF International Conference on Computer Vision 2021, Montreal, BC, Canada, 11–17 October 2021; pp. 3303–3312.
47. Liu, Y.; Wen, Y.; Peng, S.; Lin, C.; Long, X.; Komura, T.; Wang, W. Gen6D: Generalizable model-free 6-DoF object pose estimation from RGB images. In Proceedings of the European Conference on Computer Vision (ECCV) 2022, Tel Aviv, Israel, 23–27 October 2022; pp. 298–315.

48. Su, Y.; Saleh, M.; Fetzer, T.; Rambach, J.; Navab, N.; Busam, B.; Stricker, D.; Tombari, F. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2022, New Orleans, LA, USA, 18–24 June 2022; pp. 6738–6748.
49. Lian, R.; Ling, H. Checkerpose: Progressive dense keypoint localization for object pose estimation with graph neural network. In Proceedings of the IEEE/CVF International Conference on Computer Vision 2023, Paris, France, 2–6 October 2023; pp. 14022–14033.
50. Wen, B.; Yang, W.; Kautz, J.; Birchfield, S. FoundationPose: Unified 6D pose estimation and tracking of novel objects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2024, Seattle, WA, USA, 16–22 June 2024; pp. 17868–17879.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.