

# Vote-based RGBD Fusion for Hand-held Object Pose Estimation

---

## ARTICLE INFO

### Keywords:

Object pose estimation  
Object recognition  
Multimodal fusion

## ABSTRACT

Estimating the 6D pose of objects in hand is a challenging and critical problem in computer vision and robotics, particularly for applications like robotic manipulation, human-robot interaction, and augmented reality (AR). Leveraging multi-modal data, such as color (RGB) and depth (D) images, offers a promising approach. Despite significant progress in representation learning across these modalities, this field faces two major issues. First, the presence of a hand often causes significant occlusions, which conventional object pose estimation methods struggle to handle effectively. Second, existing approaches typically extract features from two separate backbones and fuse the data at the feature level. This fusion mechanism can cause a representation distribution shift and potential disruption during fine-tuning due to dense interactions between the RGB and depth branches. In this work, we propose a novel deep neural network for hand-held object pose estimation using an RGB-D image as input. Instead of fusing data at the feature level, we introduce a vote-based fusion mechanism to effectively combine multimodal data for the pose estimation task, particularly for occluded objects in the presence of a hand. Additionally, we model the relationship between the hand and the object through keypoint interaction, incorporating this relationship into the object pose estimation process. Through experiments on three public datasets, our approach demonstrates significant improvements in accuracy and robustness over existing methods, achieving an accuracy improvement of up to 15%.

---

## 1. Introduction

Estimating the 6D pose of hand-held objects is an important yet challenging task in robotics (Pfanne et al. (2018); Anzai and Takahashi (2020); Okafor et al. (2024)), virtual reality Muhanna (2015), among others. It plays a significant role in applications such as robotic manipulation or human-robot interaction (Andrychowicz et al. (2020); Handa et al. (2020)). The increasing availability of 3D sensors has made RGB-D data more accessible, which, when incorporating 3D geometric information from depth sensors, allows for more accurate and robust 6D pose estimation compared to using RGB data alone. However, despite notable advancements in multi-modal representation learning, significant challenges remain. These challenges are primarily due to occlusions caused by the hand and the complexities involved in effectively fusing RGB and depth data (Chao et al. (2021)). Traditional methods (Wang et al. (2021b); Peng et al. (2019); Wang et al. (2019); Sharma and Mir (2022)) struggle with accuracy drops when hands occlude objects, as hands obscure critical features needed for precise pose estimation. These occlusions introduce ambiguities and complexities, making it difficult to determine the object's orientation and position. Some approaches attempt to segment the hand or predict occluded regions, but these add complexity and often fall short under challenging conditions (He et al. (2020); Castro and Kim (2023)). Additionally, hand-object interactions introduce non-rigid transformations, further complicating pose estimation by distorting the object's perceived shape and pose due to varying grips and manipulations.

Current mainstream RGBD fusion methods typically employ separate RGB pre-trained backbones to extract features from RGB images and depth maps (Wang et al. (2019); He et al. (2020, 2021)). These backbones independently extract features from their respective modalities, which are then integrated during subsequent processing stages. Despite achieving high performance on benchmark datasets, several critical issues persist. The RGBD backbones are designed to handle input as image-depth pairs, which contrasts with the single-image input used in RGB pretraining. This difference often leads to a significant shift in representation distribution. Specifically, the features learned from single RGB images may not directly align with those extracted from RGBD inputs, impacting the model's ability to generalize effectively across both modalities. Moreover, during fine-tuning, extensive interactions occur between the RGB and depth branches. These interactions are crucial for integrating complementary information from both sources. However, the dense coupling between these branches can potentially disturb the original representation distribution learned by the pretrained RGB backbone. This phenomenon may hinder the model's capacity to leverage the full potential of the RGB features, thereby affecting overall performance and generalization ability on RGBD tasks.

---

ORCID(s):

In this paper, we present a novel deep neural network specifically designed for hand-held object pose estimation using RGB-D images. Different from traditional feature-level fusion methods, our approach introduces a vote-based fusion mechanism that effectively integrates multimodal data. This mechanism leverages a voting scheme where both 2D and 3D keypoints cast votes for the object's pose, particularly enhancing the estimation in scenarios with occluded objects. By combining votes from different modalities, our method mitigates the issues of representation distribution shift and potential disruptions during fine-tuning, leading to more accurate pose predictions. Furthermore, we model the interaction between the hand and the object through self-attention mechanism, integrating this relationship into the pose estimation process. Our network predicts keypoints for both the hand and the object, using these keypoints to establish a contextual relationship that enhances the pose estimation, especially when the object is partially obscured by the hand. This hand-object keypoint interaction modeling allows our method to account for non-rigid transformations and occlusions introduced by the hand, significantly improving the robustness of the pose estimation.

The primary contributions of this work can be summarized as follows:

- **Vote-based Fusion Module  $\mathcal{M}_{fus}$ :** We introduce a vote-based fusion mechanism that integrates RGB and depth data more effectively than traditional feature-level fusion methods. This module utilizes both 2D and 3D features, leveraging a radius-based neighborhood projection and channel attention mechanisms to enhance the representation of crucial features while mitigating noise and redundancy.
- **Hand-aware Object Pose Estimation Module  $\mathcal{M}_{hao}$ :** By modeling the relationship between the hand and the object through a self-attention mechanism, we enhance the pose estimation process. This self-attention mechanism captures complex spatial relationships between keypoints, which is crucial for accurately estimating the pose of objects that are partially occluded by the hand.
- **Comprehensive Experimental Validation:** We provide extensive experimental validation across three public datasets: DexYCB ([Chao et al. \(2021\)](#)), FPHAB ([Garcia-Hernando et al. \(2018\)](#)), and HO-3D ([Hampali et al. \(2020\)](#)). Our method consistently outperforms state-of-the-art approaches, achieving significant improvements in all metrics. Additionally, our method achieves competitive inference times, making it suitable for real-world applications.

## 2. Related work

### 2.1. Object Pose Estimation From Visual Inputs

Object pose estimation is a well-explored field in computer vision, focusing on determining the position and orientation (pose) of objects from visual inputs such as RGB or depth images ([Wang et al. \(2019\)](#); [He et al. \(2020, 2021\)](#)). Deep learning-based methods have significantly advanced pose estimation by learning rich feature representations directly from the data ([Wang et al. \(2021a\)](#); [Peng et al. \(2019\)](#)). These methods can be broadly categorized into two main types: depth-based methods, RGB-based methods, RGBD-based methods. Depth-based object pose estimation uses depth sensors to capture the 3D structure of objects, extracting geometric information from depth maps ([Wang et al. \(2021b\)](#); [Gao et al. \(2020\)](#); [Guo et al. \(2021\)](#)). This approach is beneficial in low-light conditions or when RGB-based methods fail due to insufficient texture. However, depth sensors can produce noisy or incomplete data, especially with occlusions or reflective surfaces, and processing dense point clouds can be computationally expensive for real-time applications. RGB-based methods estimate object poses using color images, leveraging texture and color information to identify and localize objects ([Billings and Johnson-Roberson \(2019\)](#); [Peng et al. \(2019\)](#); [Wang et al. \(2021a\)](#)). Common techniques include keypoint detection, feature matching, and deep learning-based regression. These methods perform well in well-lit environments with distinct textures but struggle with varying lighting conditions and occlusions. They are also less effective for objects with uniform or repetitive textures, where the lack of distinct features makes pose estimation challenging.

RGBD-based methods combine the strengths of both RGB and depth-based approaches by leveraging multimodal data ([Wang et al. \(2019\)](#); [He et al. \(2020\)](#); [Hong et al. \(2024\)](#)). These methods utilize both color and depth information to provide a more comprehensive understanding of the object's geometry and appearance. By integrating RGB and depth data, RGBD-based methods can achieve higher accuracy and robustness in pose estimation. Existing data fusion methods often involve combining features from RGB and depth modalities at various stages of the neural network.

Common approaches include early fusion (combining raw data), middle fusion (combining intermediate features), and late fusion (combining final predictions). Each method has its advantages and drawbacks. Early fusion can lead to high-dimensional inputs and increased computational costs, while late fusion may miss out on capturing interactions between modalities during feature extraction. Middle fusion strikes a balance but can still suffer from representation distribution shifts. In hand-held object pose estimation, the presence of the hand introduces non-rigid transformations and occlusions that further complicate the fusion process. Existing object pose methods may not effectively model the interaction between the hand and the object, leading to reduced accuracy in pose estimation.

Another line of work focuses on joint hand and object pose estimation (Doosti et al. (2020); Lin et al. (2023); Wang et al. (2023); Woo et al. (2023)). Most of these methods use RGB as input and are typically based on parameterized hand models, such as MANO (Romero et al. (2022)). Estimating both hand and object poses is inherently complex due to the high degree of articulation in the hand, self-occlusions, and the intricate interactions between the hand and the object. These challenges make the estimation process more error-prone, particularly when precise localization of the object is critical. Moreover, hand-object interaction introduces non-rigid transformations that are difficult to model accurately. Despite these challenges, joint hand-object estimation is useful in scenarios such as virtual reality (VR) and surgical robotics, where both hand and object poses are critical for precise control. However, in many hand-held object scenarios, the primary focus is on object pose estimation. For example, in inventory scanning and mobile AR applications, the main goal is to determine the object's position and orientation to overlay information or capture barcodes. In these cases, while the hand manipulates the object, the accuracy of the object's pose is far more crucial than that of the hand. Therefore, simplifying the problem to focus on object pose estimation can lead to more practical and efficient solutions for these applications.

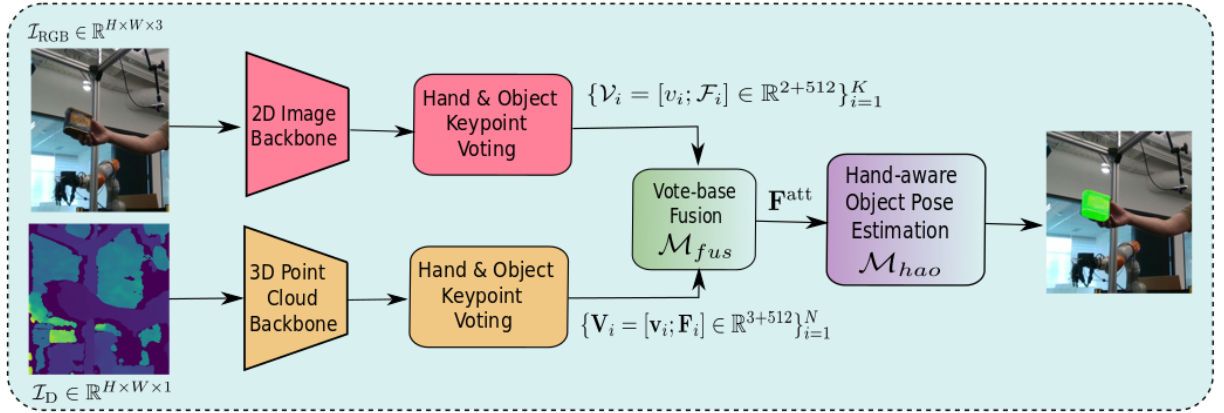
In this work, we concentrate on hand-held object pose estimation, where the focus is primarily on the object's pose rather than the hand's. We propose a novel deep neural network that utilizes RGB-D images to estimate the pose of hand-held objects. Instead of traditional feature-level fusion methods, our approach introduces a vote-based fusion mechanism that combines multimodal data more effectively. This method is particularly advantageous in scenarios where occlusions caused by the hand are present, as it enhances the robustness and accuracy of object pose estimation by leveraging both RGB and depth information in a more synergistic manner.

## 2.2. Voting Mechanism In Visual Tasks

Voting mechanisms have been widely used in computer vision tasks to enhance robustness and accuracy. These mechanisms involve aggregating multiple hypotheses or predictions to arrive at a consensus, which is particularly useful in noisy or ambiguous scenarios. The use of voting in visual tasks can be traced back to classical techniques like the Hough Transform, which aggregates votes in a parameter space to detect shapes in images (Hough (1959)). In recent years, voting mechanisms have been integrated into deep learning frameworks to improve tasks such as object detection or pose estimation. VoteNet (Qi et al. (2019)), for instance, employs a voting scheme where 3D points cast votes for object centers, leading to robust object detection in point clouds. Inspired by this, Xie et al. (2021) developed an enhanced VoteNet-based detector for cluttered indoor scenes, replacing the traditional MLP with an Attentive MLP (AMLP) for better feature description. Further, MlcVNet (Xie et al. (2020)) introduced Multi-Level Context VoteNet (MLCVNet), integrating context modules into voting and classification to encode contextual information at multiple levels. These include the Patch-to-Patch Context (PPC) for point patches, Object-to-Object Context (OOC) for object candidates, and Global Scene Context (GSC) for the overall scene. In RGB object pose estimation, voting mechanisms have proven highly effective. Peng et al. (2019) introduced PVNet, which uses pixel-wise vector regression to vote for keypoint locations, aiding in the localization of occluded or truncated keypoints. He et al. (2020) extended this by detecting 3D keypoints and estimating 6D pose parameters through a deep Hough voting network. Di et al. (2022) later proposed GPV-Pose, which leverages geometric constraints for category-level pose estimation, using a 3D graph convolution encoder with branches for pose regression, symmetry-aware reconstruction, and point-wise bounding box voting.

Inspired by these successful implementations of voting mechanisms, we introduce a novel vote-based fusion mechanism for RGB-D images. Unlike existing methods that generate votes from depth images, RGB images, or fused RGB-D features, our approach separately performs voting in the 2D image branch and the 3D point cloud branch before combining the results. By leveraging both 2D and 3D keypoints, this method integrates multimodal data more effectively than traditional feature-level fusion methods, significantly enhancing pose estimation accuracy, especially in scenarios involving occluded objects.

### 3. Methodology



**Figure 1:** Overview of our proposed framework for estimating the 6D pose of hand-held objects from RGB-D images. The framework comprises several key components: (1) feature extraction backbones for both 2D images and 3D point clouds, which process the RGB and depth information, respectively; (2) voting modules that generate votes for keypoint locations in both 2D and 3D spaces; (3) a vote-based fusion module  $\mathcal{M}_{fus}$  that effectively combines the multimodal data to address the challenges of occlusions and representation distribution shifts; and (4) a hand-aware object pose estimation module  $\mathcal{M}_{hao}$ , which models the interactions between the hand and the object using a self-attention mechanism.

Our objective is to determine the 6D pose of a hand-held object within an RGB-D image. We assume the availability of an accurate 3D model of the object, with its coordinate system  $\mathcal{O}$  defined in the model's 3D space. The pose of the object is described by a rigid transformation from the object coordinate system to the camera coordinate system  $\mathcal{G}$ . This transformation comprises a rotation matrix  $R \in SO(3)$  and a translation vector  $t \in \mathbb{R}^3$ , collectively represented as  $\xi = [R|t]$ . Figure 1 illustrates the overall pipeline of our framework. Our network consists of backbones for extracting features from both 2D images and 3D point clouds, voting modules, a vote-based fusion module  $\mathcal{M}_{fus}$ , and a hand-aware object pose estimation module  $\mathcal{M}_{hao}$ .

#### 3.1. 3D Point Cloud Branch

Given a depth image, the 3D point cloud branch network converts the image to a 3D point cloud and proceeds to cast votes for 3D keypoints of both hand and objects. Inspired by VoteNet (Qi et al. (2019)), wherein 3D points participate in voting for object centers in object detection tasks, we have formulated our network to predict keypoint locations for individual 3D points. Specifically, we utilize the PointNet++ architecture (Qi et al. (2017)) with multi-scale grouping as our backbone network to extract geometric features. The backbone network takes  $N$  points as input and enriches them with high-dimensional features  $\{\mathbf{p}_i\}_{i=1}^N$  where  $\mathbf{p}_i = [\mathbf{x}_i; \mathbf{f}_i]$  with  $\mathbf{x}_i \in \mathbb{R}^3$  being the point location in 3D space and  $\mathbf{f}_i \in \mathbb{R}^{512}$  being a feature vector.

Subsequently, the points  $\{\mathbf{p}_i\}_{i=1}^N$  are fed into a multi-layer perceptron (MLP) to compute votes  $\{\mathbf{V}_i = [\mathbf{v}_i; \mathbf{F}_i] \in \mathbb{R}^{3+512}\}_{i=1}^N$ . The MLP consists of four fully connected layers, ReLU and batch normalization. Each vote  $\mathbf{V}_i$  is represented by a point  $\mathbf{v}_i$  in 3D space with its Euclidean coordinates supervised to be close to a keypoint  $\mathbf{s}_i$ , and a feature vector  $\mathbf{F}_i$ . To supervise the learning of votes  $\{\mathbf{V}_i = [\mathbf{v}_i; \mathbf{F}_i] \in \mathbb{R}^{3+512}\}_{i=1}^N$ , we apply a regression loss:

$$L_{3d-vote} = \frac{1}{N_{oh}} \sum_i \|\mathbf{v}_i - \hat{\mathbf{v}}_i\|_H \cdot \mathbb{1}(\mathbf{v}_i) \quad (1)$$

Here,  $N_{oh}$  is the count of the total number of points on the object and hand,  $\hat{\mathbf{v}}_i$  is the location of the closest ground truth keypoint,  $\|\cdot\|_H$  is the Huber norm and  $\mathbb{1}(\cdot)$  is a binary function indicating whether the point  $\mathbf{p}_i$  belongs to the hand or object. At this stage, we have a set of votes  $\{\mathbf{V}_i = [\mathbf{v}_i; \mathbf{F}_i] \in \mathbb{R}^{3+512}\}_{i=1}^N$ .

#### 3.2. 2D Image Branch

Given an input color image, this network branch predicts hand and object 2D keypoints. We adopt a pixel-wise direction prediction strategy, which encourages the network to prioritize local hand and object characteristics while

minimizing the influence of cluttered backgrounds. We employ ResNet-34 (blocks 1, 2, 3, and 4) as the encoder, coupled with four up-sampling layers serving as the decoder (He et al. (2016)). The upsampling layers consist of transposed convolutional layers with batch normalization and ReLU activation. For a given pixel  $p$ , the network predicts a scale-invariant vector  $\hat{\mathbf{u}}(p)$ , representing the direction from pixel  $p$  to the closest keypoint  $s$ .

$$\hat{\mathbf{u}}(p) = \frac{s - p}{\|s - p\|_2} \quad (2)$$

To supervise the learning of unit vectors, we apply a smooth  $L_1$  loss, similar to the approach in Girshick (2015). The loss function  $L_{2d\text{-vote}}$  is defined as:

$$L_{2d\text{-vote}} = \frac{1}{N_{oh}} \sum_{p \in (\mathcal{O} \cup \mathcal{H})} \text{smooth}_{L_1}(\hat{\mathbf{u}}(p) - \mathbf{u}(p)) \quad (3)$$

Here,  $N_{oh}$  is the count of the total number of pixels on the object and hand.  $\mathbf{u}$  denotes the predicted vector,  $\hat{\mathbf{u}}$  represents the ground truth unit vector, and  $p \in (\mathcal{O} \cup \mathcal{H})$  indicates that pixel  $p$  pertains to hand or object. Votes are generated by randomly selecting two pixels situated within a specific radius and then calculating the intersection of their respective vectors. Additionally, a representative feature is computed for these selected pixels through feature concatenation. This process is iterated  $K$  times, leading to the generation of votes  $\{\mathcal{V}_i = [v_i; \mathcal{F}_i] \in \mathbb{R}^{2+512}\}_{i=1}^K$ .  $v_i \in \mathbb{R}^2$  represents the 2D vote location, and  $\mathcal{F}_i \in \mathbb{R}^{512}$  represents the high-dimensional feature of the vote  $v_i$ .

### 3.3. Vote-based Feature Fusion

Given the votes  $\{\mathcal{V}_i = [v_i; \mathcal{F}_i] \in \mathbb{R}^{3+512}\}_{i=1}^N$  from the 3D branch and  $\{\mathcal{V}_i = [v_i; \mathcal{F}_i] \in \mathbb{R}^{2+512}\}_{i=1}^K$  from the 2D branch, we perform vote-based fusion using a channel attention mechanism adapted for 3D data. For each geometric vote feature  $\mathbf{F}_i$  paired with its 3D point coordinate  $\mathbf{v}_i$ , we retrieve the corresponding visual vote features from  $\mathcal{F}$  by projecting its neighborhood within a radius  $r$  onto the image plane. We then sample the  $k$  nearest neighbor votes within this region to gather their visual features. If fewer than  $k$  pixels exist in the corresponding region, null features are padded. The collected visual features are integrated using max pooling and processed through Multi-Layer Perceptrons (MLPs) to align their channel size with that of the point cloud features, producing modified visual vote features  $\mathcal{F}'$ :

$$\mathcal{F}_i^{\text{proj}} = \text{MaxPool}(\{\mathcal{F}_j \mid \|v_j - \mathbf{v}_i\|_2 < r\})$$

$$\mathcal{F}'_i = \text{MLP}_{\text{match}}(\mathcal{F}_i^{\text{proj}})$$

Next, these integrated visual vote features  $\mathcal{F}'_i$  are concatenated with the geometric vote features  $\mathbf{F}_i$ , and a shared MLP is applied to produce fused vote features  $\{\mathbf{V}_i = [\mathbf{v}_i; \mathbf{F}_i^{\text{fus}}] \in \mathbb{R}^{3+1024}\}_{i=1}^N$ :

$$\mathbf{F}_i^{\text{fus}} = \text{MLP}_{\text{shared}}([\mathbf{F}_i; \mathcal{F}'_i])$$

To enhance these fused features, we employ a 3D channel attention module. This module uses global average pooling across all votes to generate a channel descriptor for each feature map. Specifically, global average pooling compresses each channel by computing the average of all values in that channel across all votes, resulting in a descriptor vector  $\mathbf{z}_c$ :

$$\mathbf{z}_c = \frac{1}{N} \sum_{i=1}^N \mathbf{F}_i^{\text{fus}}$$

The channel descriptors are then passed through two fully connected layers to learn a weight vector. The first fully connected layer reduces the dimensionality, and the second fully connected layer restores it back to the original number of channels. Both layers use ReLU activations, except the final layer, which uses a sigmoid activation to normalize the weights between 0 and 1:

$$\mathbf{w} = \sigma(\text{FC}_2(\text{ReLU}(\text{FC}_1(\mathbf{z}))))$$

where  $\sigma$  denotes the sigmoid activation function. The learned weights are used to scale the channels of the fused feature map, effectively emphasizing the most informative channels and suppressing the less relevant ones:



$$\mathbf{F}_i^{\text{att}} = \mathbf{w} \odot \mathbf{F}_i^{\text{fus}}$$

where  $\odot$  denotes the Hadamard product (element-wise multiplication). This attention-enhanced fused feature,  $\mathbf{F}_i^{\text{att}}$ , leverages both the geometric and visual information for improved hand-held object pose estimation. The channel attention mechanism recalibrates the features in each channel of the original feature points, enhancing the representation of more informative channels and suppressing less relevant ones. The design of this vote-based feature fusion is well-suited for RGB-D data fusion. By combining 2D and 3D features, we leverage the texture and color of RGB images with the spatial details of depth data. The radius-based neighborhood projection keeps the integration locally relevant, capturing necessary fine-grained details. Additionally, the channel attention mechanism enhances the fused features by prioritizing the most informative channels, reducing noise and redundancy, and dynamically adjusting each channel's contribution based on its relevance.

### 3.4. Hand-aware Object Pose Estimation

Given a set of votes  $\{\mathbf{V}_i = [\mathbf{v}_i; \mathbf{F}_i^{\text{att}}] \in \mathbb{R}^{3+512}\}_{i=1}^N$ , we sample a subset of  $M$  votes (where  $M < N$ ) using farthest point sampling based on  $\{\mathbf{v}_i\}$  in 3D Euclidean space, to get  $\{\mathbf{v}_i^m\}$  with  $m = 1, \dots, M$ . Then we form  $M$  clusters by finding neighboring votes to each of the  $\mathbf{v}_i^m$ 's 3D location:

$$\mathbf{C}_m = \{\mathbf{V}_i \mid \|\mathbf{v}_i - \mathbf{v}_i^m\| \leq r\} \quad \text{for } m = 1, \dots, M$$

Here,  $\mathbf{C}_m$  is the set of votes in the  $m$ -th cluster. Votes from each cluster are independently processed by  $\text{MLP}_1$  before being max-pooled (channel-wise) to a single feature vector and passed to  $\text{MLP}_2$  where information from different votes is further combined. Finally, each cluster yields a keypoint with a 3D position  $\mathbf{x}_j^{\text{kp}} \in \mathbb{R}^3$  and an aggregated feature vector  $F_i \in \mathbb{R}^{512}$ .

To better learn the relationship between the hand and the object, we integrate a self-attention module into the keypoint feature learning process. This module, inspired by the self-attention mechanism (Zhang et al. (2019)), encodes meaningful spatial relationships between the keypoints' features. Given the aggregated features  $\{F_i \in \mathbb{R}^{512}\}_{i=1}^M$  from the  $M$  keypoints, we apply the self-attention mechanism as follows. Firstly, the feature map  $F_i$  is fed into two MLPs to generate two new feature maps  $X$  and  $Y$ :

$$X = \text{MLP}_1(F_i)$$

$$Y = \text{MLP}_2(F_i)$$

Next, the attention map  $W$  is computed as:

$$W_{j,i} = \frac{\exp(Y_j \cdot X_i^T)}{\sum_{i=1}^M \exp(Y_j \cdot X_i^T)}$$

where  $W_{j,i}$  indicates that the  $i$ -th keypoint impacts the  $j$ -th keypoint. The original feature map  $F_i$  is then fed into another MLP to output a new feature map  $Z$ :

$$Z = \text{MLP}_3(F_i)$$

We multiply  $Z$  with the transpose of  $W$  to generate the aggregated features  $A^P$ :

$$A^P = W^T Z$$

We then add a scale parameter  $\mu$  and the original features to obtain the final enhanced feature map  $F_i^{\text{att}}$ :

$$F_i^e = \mu A^P + F_i = \mu W^T Z + F_i$$

This self-attention mechanism allows the network to learn the dependencies between different keypoints, capturing the complex relationships between the hand and the object, thus enhancing the feature representation for the hand-held object pose estimation. By incorporating this self-attention mechanism, the model can effectively leverage the spatial relationships between the keypoints to improve the accuracy and robustness of the pose estimation, leading to a more precise understanding of the hand-object interaction.

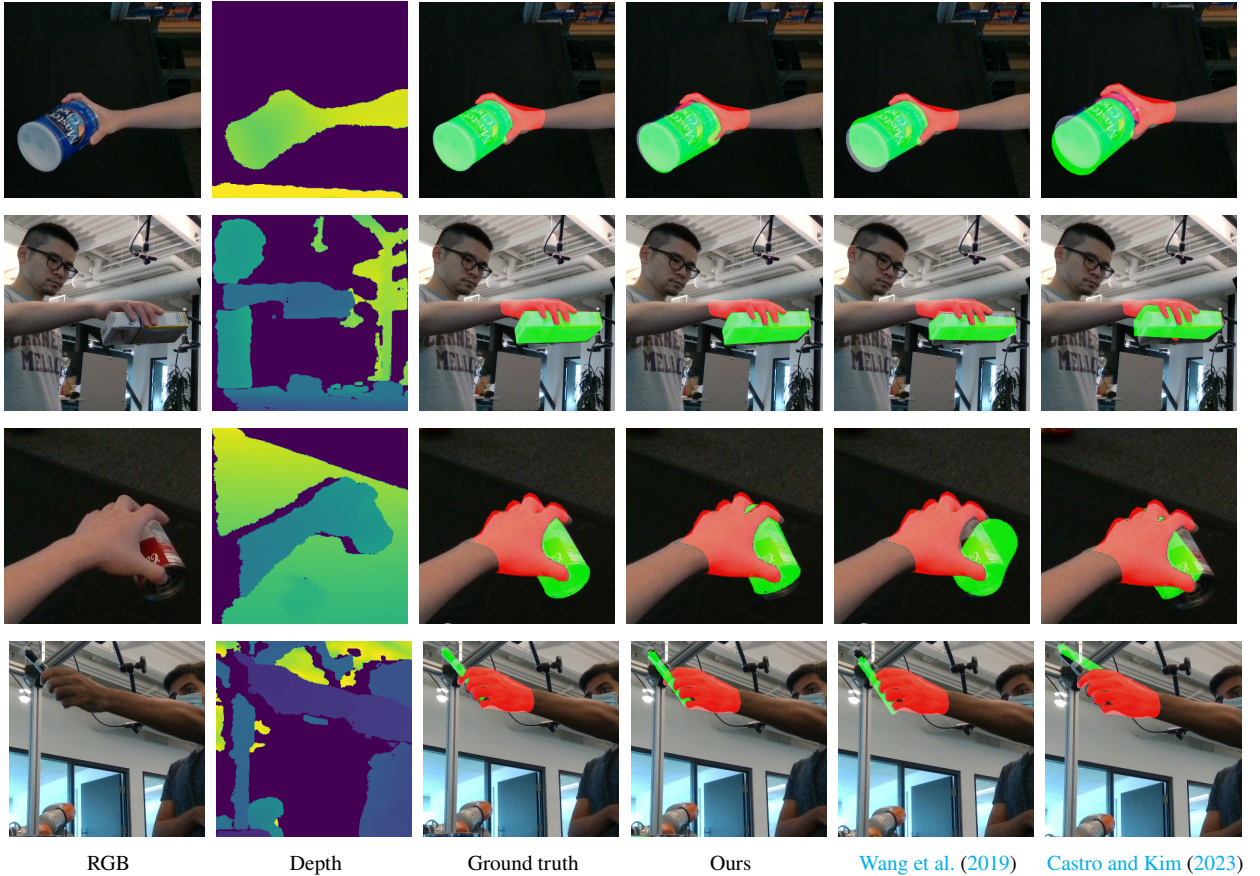
Given the enhanced keypoint features  $\{\mathbf{F}_i^e \in \mathbb{R}^{512}\}_{i=1}^M$ , we proceed to regress the object pose. Each enhanced feature  $\mathbf{F}_i^e$  is input into an MLP-based regression network to predict the 3D object pose parameters. This regression network consists of several fully connected layers with ReLU activations, followed by an output layer that predicts the translation vector  $\mathbf{t} \in \mathbb{R}^3$  and the rotation quaternion  $\mathbf{q} \in \mathbb{R}^4$ . The predicted pose  $\{\mathbf{t}, \mathbf{q}\}$  represents the 3D translation and rotation of the object relative to the hand.

To train the network, we jointly supervise the learning of our modules using a multi-task loss function. This multi-task loss integrates several task-specific losses, each weighted by a respective factor, to ensure a balanced learning process across different aspects of pose estimation.

$$L = L_{pose} + L_{3d-vote} + L_{2d-vote} + L_{sem} \quad (4)$$

Here,  $L$  represents the total loss, composed of the pose loss ( $L_{pose}$ ), 3D voting loss ( $L_{3d-vote}$ ), 2D voting loss ( $L_{2d-vote}$ ), and semantic classification loss ( $L_{sem}$ ). Following Wang et al. (2019), the pose loss is calculated as the distance between points sampled on the object model in the ground truth pose and the corresponding points transformed by the predicted pose. For symmetric objects, we minimize the distance from each point on the predicted model orientation to the nearest point on the ground truth model.

## 4. Evaluation



**Figure 2:** Qualitative results. (a) and (b) are the input RGBD images. (c) shows the rendered images using ground truth hand and object poses. (d), (e), and (f) display the rendered images using ground truth hand poses and object poses predicted by our method, Wang et al. (2019), and Castro and Kim (2023), respectively.

We evaluate our proposed approach on three publicly available datasets: DexYCB (Chao et al. (2021)), FPHAB (Garcia-Hernando et al. (2018)), and HO-3D (Hampali et al. (2020)). These datasets are chosen for their comprehensive representation of hand-held object interactions, detailed annotations, and diverse scenarios, making them ideal for benchmarking object pose estimation. This evaluation compares our method with state-of-the-art object pose estimation techniques. Furthermore, we assess the impact of our RGB-D fusion module by substituting alternative fusion methods and analyzing the accuracy improvements.

#### 4.1. Datasets

**DexYCB Dataset** (Chao et al. (2021)). This dataset provides 582,000 RGB-D frames across 1,000 sequences with 10 subjects interacting with 20 different objects. Utilizing eight RGB-D cameras, it captures interactions from multiple angles, ensuring rich data diversity. The dataset's precise 3D annotations of both hands and objects make it highly suitable for evaluating hand-held object pose estimation in cluttered environments.

**FPHAB Dataset** (Garcia-Hernando et al. (2018)). With over 100,000 frames, this dataset features 45 hand-object interaction categories involving 26 objects. Collected using a motion capture system, it offers detailed 3D annotations, crucial for testing pose estimation models under diverse hand configurations and object manipulations. This diversity aids in assessing the robustness of pose estimation techniques in realistic scenarios.

**HO-3D Dataset** (Hampali et al. (2020)). Comprising 77,558 frames across 68 sequences, this dataset focuses on challenging hand-object interactions with significant occlusions. It includes RGB images with detailed 3D annotations of hand-object poses, enabling the evaluation of pose estimation methods in complex scenarios. The dataset's challenging conditions test the effectiveness of our approach in real-world applications.

#### 4.2. Implementation Details.

All images from each dataset are cropped and resized to  $256 \times 256$  pixels. Our implementation<sup>1</sup> uses a pre-trained ResNet34 model, originally trained on ImageNet, as the encoder for RGB images. For point cloud feature extraction, we randomly sample 12,288 points from depth images and use a PointNet++-based feature learning network (Qi et al. (2017)), which includes 4 set abstraction (SA) layers and 2 feature propagation (FP) layers. Following FFB6D (He et al. (2021)), we select 256 hand and object keypoints using SIFT-FPS algorithm. The entire system is implemented using PyTorch and Python, running on an NVIDIA RTX-3090 GPU with an Intel Xeon CPU (12 cores, 2.1GHz), utilizing CUDA and the Linux operating system. For all datasets, we use the Adam optimizer with an initial learning rate of 0.01. The learning rate decays by 0.1 at epochs 80, 140, and 200. We train the model with a batch size of 8 and apply standard data augmentation techniques.

#### 4.3. Evaluation metric

To assess the accuracy of the estimated pose  $\hat{P}$  relative to the ground-truth pose  $\bar{P}$  of an object model  $M$ , we use the widely adopted Average Distance of Model Points (ADD) metric (Hinterstoisser et al. (2012)). This metric calculates the average distance between corresponding vertices of the object model in the ground-truth pose and the estimated pose. Formally, given the ground truth rotation  $\bar{R}$  and translation  $\bar{t}$ , and the estimated rotation  $\hat{R}$  and translation  $\hat{t}$ , the ADD is defined as:

$$ADD = \frac{1}{m} \sum_{x \in M} \| (\bar{R}x + \bar{t} - \hat{R}x + \hat{t}) \| \quad (5)$$

where  $m$  is the number of model points. For symmetric objects, we adapt the metric by computing the average distance using the closest point distance method, as in Brégier et al. (2017). We evaluate the prediction accuracy using average precision (AP) (Brégier et al. (2017)). A 6D pose estimate is classified as a true positive if the average distance is less than 10% of the diameter of the smallest bounding sphere of the object. Additionally, we report the area under the ADD curve (AUC) (Wang et al. (2019)), providing a measure of pose estimation performance across varying thresholds.

#### 4.4. Results

Table 1 and 2 present the quantitative results, while Figure 2 illustrates the qualitative results. The proposed method achieves the highest accuracy across all datasets in both AUC and AP metrics. For instance, in the DexYCB dataset, our

<sup>1</sup>Our code and other materials are available at [https://github.com/papersubm/6d\\_object\\_inhand](https://github.com/papersubm/6d_object_inhand)



**Table 1**

Quantitative results on the DexYCB (Chao et al. (2021)), FPHAB (Garcia-Hernando et al. (2018)), and HO-3D (Hampali et al. (2020)) datasets without Iterative Refinement. Depth-based methods (Wang et al. (2021b); Gao et al. (2020); Guo et al. (2021)), RGB methods (Billings and Johnson-Roberson (2019); Peng et al. (2019); Wang et al. (2021a); Castro and Kim (2023)), and RGBD methods (Wang et al. (2019); He et al. (2020, 2021); Wu et al. (2023); Hong et al. (2024); Lin et al. (2024)) are compared with our proposed method (Ours). The runtime for each method is measured in milliseconds (ms).

Method	DexYCB		FPHAB		HO-3D		Time
	<i>AUC</i>	<i>AP</i>	<i>AUC</i>	<i>AP</i>	<i>AUC</i>	<i>AP</i>	
Wang et al. (2021b)	52.3	53.4	54.2	55.1	53.6	54.7	42
Gao et al. (2020)	55.4	56.1	57.3	58.2	56.7	57.6	36
Guo et al. (2021)	57.1	58.0	59.0	59.9	58.4	59.3	33
Billings and Johnson-Roberson (2019)	54.3	55.5	56.2	57.1	55.6	56.9	35
Peng et al. (2019)	57.2	58.1	59.1	60.0	58.5	59.4	38
Wang et al. (2021a)	59.6	60.5	61.5	62.4	60.9	61.8	32
Castro and Kim (2023)	61.2	62.1	63.1	64.0	62.5	63.4	30
Wang et al. (2019)	60.3	61.1	62.2	63.1	61.6	62.5	42
He et al. (2020)	62.1	63.0	64.1	65.0	63.5	64.4	49
He et al. (2021)	63.4	64.2	65.3	66.2	64.8	65.7	51
Wu et al. (2023)	65.0	65.8	66.9	67.8	66.4	67.3	46
Hong et al. (2024)	66.3	67.1	68.2	69.1	67.6	68.5	55
Lin et al. (2024)	67.5	68.4	69.6	70.5	68.9	69.8	65
Ours	<b>80.3</b>	<b>81.2</b>	<b>82.1</b>	<b>83.4</b>	<b>81.5</b>	<b>82.6</b>	40

**Table 2**

Quantitative results on the DexYCB (Chao et al. (2021)), FPHAB (Garcia-Hernando et al. (2018)), and HO-3D (Hampali et al. (2020)) datasets with Iterative Refinement. The runtime for each method is measured in milliseconds (ms).

Method	DexYCB		FPHAB		HO-3D		Time
	<i>AUC</i>	<i>AP</i>	<i>AUC</i>	<i>AP</i>	<i>AUC</i>	<i>AP</i>	
Wang et al. (2021b)	61.5	62.4	63.2	64.1	62.7	63.6	163
Gao et al. (2020)	63.2	64.0	65.1	66.0	64.5	65.3	187
Guo et al. (2021)	64.9	65.7	66.6	67.5	66.0	66.9	165
Billings and Johnson-Roberson (2019)	62.8	63.7	64.6	65.5	63.9	64.8	160
Peng et al. (2019)	65.2	66.1	67.1	68.0	66.5	67.4	170
Wang et al. (2021a)	66.9	67.8	68.7	69.6	68.2	69.1	185
Castro and Kim (2023)	68.3	69.2	70.2	71.1	69.6	70.5	178
Wang et al. (2019)	67.0	67.8	68.8	69.7	68.3	69.1	240
He et al. (2020)	68.5	69.4	70.4	71.3	69.9	70.7	270
He et al. (2021)	69.8	70.7	71.7	72.6	71.2	72.0	288
Wu et al. (2023)	71.2	72.1	73.1	74.0	72.6	73.4	215
Hong et al. (2024)	72.3	73.2	74.2	75.1	73.7	74.5	265
Lin et al. (2024)	73.5	74.4	75.4	76.3	74.9	75.7	290
Ours	<b>86.7</b>	<b>87.2</b>	<b>88.0</b>	<b>88.5</b>	<b>88.3</b>	<b>87.7</b>	200

method achieves an AUC of 80.3 and an AP of 81.2 without iterative refinement, and an AUC of 86.7 and an AP of 87.2 with iterative refinement. The results indicate that RGBD-based methods generally outperform RGB-only (Billings and Johnson-Roberson (2019); Peng et al. (2019); Wang et al. (2021a); Castro and Kim (2023)) and depth-only methods (Wang et al. (2021b); Gao et al. (2020); Guo et al. (2021)), leveraging the complementary information from both modalities for superior 3D pose estimation. Among the RGBD methods (Wang et al. (2019); He et al. (2020, 2021); Wu et al. (2023); Hong et al. (2024); Lin et al. (2024)), our approach stands out not only for its accuracy but also for its

**Table 3**

Performance comparison of our method with other state-of-the-art hand-object pose estimation methods. The upper section shows results without iterative refinement, and the lower section includes iterative refinement. The runtime for each method is measured in milliseconds (ms).

	DexYCB		FPHAB		HO-3D		Time
Method	<i>AUC</i>	<i>AP</i>	<i>AUC</i>	<i>AP</i>	<i>AUC</i>	<i>AP</i>	<i>ms</i>
Doosti et al. (2020)	56.5	57.3	58.7	59.3	57.2	58.5	131
Lin et al. (2023)	58.7	59.5	61.8	62.1	60.0	61.1	143
Wang et al. (2023)	61.3	62.1	63.0	64.5	62.6	63.4	154
Qi et al. (2024)	63.0	64.5	65.4	66.3	64.1	65.2	120
Ours	<b>80.3</b>	<b>81.2</b>	<b>82.1</b>	<b>83.4</b>	<b>81.5</b>	<b>82.6</b>	40
Doosti et al. (2020)	65.2	67.1	66.2	67.0	64.3	67.0	461
Lin et al. (2023)	69.0	71.2	69.0	70.7	68.1	69.0	496
Wang et al. (2023)	69.3	70.4	71.2	73.8	70.1	71.8	535
Qi et al. (2024)	70.7	72.6	73.0	75.3	72.4	74.2	422
Ours	<b>86.7</b>	<b>87.2</b>	<b>88.0</b>	<b>88.5</b>	<b>88.3</b>	<b>87.7</b>	200

computational efficiency. Despite not being the fastest overall, our method is the fastest among RGBD methods, with a runtime of 40 ms without iterative refinement and 200 ms with iterative refinement, making it feasible for real-world applications. Depth-based methods, such as those by Wang et al. (2021b), Gao et al. (2020), and Guo et al. (2021), exhibit reasonable speeds but do not achieve the same level of accuracy as RGBD methods. Conversely, RGB methods (Billings and Johnson-Roberson (2019); Peng et al. (2019); Wang et al. (2021a); Castro and Kim (2023)), while faster, sacrifice some accuracy compared to RGBD methods. This balance of high accuracy and competitive speed highlights the practical applicability of our method in real-world scenarios.

We further compare the proposed method with joint hand-object pose estimation approaches, as shown in Table 3. It is important to note that we evaluate the performance exclusively on object pose estimation, even though the compared methods estimate both hand and object poses. By focusing solely on object pose estimation, our method avoids the complexities inherent in modeling hand articulations. The model is simpler, with fewer parameters, making it easier to train and less prone to overfitting. This design choice results in a more streamlined and accurate estimation process, leading to significantly higher AUC and AP scores across all datasets. For instance, without iterative refinement, our method achieves an AUC of 80.3 and an AP of 81.2 on the DexYCB dataset, compared to 63.0 and 64.5, respectively, for the method by Qi et al. (2024). On the FPHAB dataset, our method obtains an AUC of 82.1 and an AP of 83.4, far outperforming Qi et al. (2024), which achieves 65.4 and 66.3. The same trend continues on the HO-3D dataset, where our method achieves an AUC of 81.5 and an AP of 82.6, compared to 64.1 and 65.2 for Qi et al. (2024). Moreover, the exclusion of hand pose estimation not only enhances accuracy but also significantly improves runtime efficiency. Our method operates at 40 ms without iterative refinement and 200 ms with iterative refinement, which is considerably faster than joint estimation methods such as Lin et al. (2023), which requires 143 ms and 496 ms. These results clearly demonstrate the advantages of simplifying the problem to focus exclusively on object pose estimation, particularly in applications where the object's pose is more critical than the hand's, such as inventory scanning and mobile AR. By concentrating on this primary task, we achieve both higher accuracy and faster performance, validating our approach as a practical and effective solution for hand-held object pose estimation in complex, occluded environments.

#### 4.5. Ablation Study

The ablation study reveals the significance of each component in our method for RGBD fusion in hand-held object pose estimation (see Table 4 and 5). Removing the hand keypoint voting mechanism resulted in a notable drop in performance across all datasets and metrics. For instance, without hand keypoints, the AUC decreased from 80.3 to 76.2 on DexYCB, 82.1 to 76.8 on FPHAB, and 81.5 to 75.2 on HO-3D. This highlights the critical role of hand keypoint voting in enhancing the accuracy of pose estimation by providing valuable spatial cues. The exclusion of the vote-based fusion module using channel attention ( $\mathcal{M}_{fus}$ ) led to the most significant performance drop, with the AUC falling to 70.5, 71.4, and 68.4 on DexYCB, FPHAB, and HO-3D, respectively. This substantial decrease underscores the importance of the vote-based fusion module in effectively combining RGB and depth features, which is crucial

**Table 4**

Ablation study without Iterative Refinement. The table compares our full method with versions that exclude hand keypoint voting (w/o hand keypoints), the vote-based fusion module using channel attention (w/o  $\mathcal{M}_{fus}$ ), and hand-aware object pose estimation using self-attention (w/o  $\mathcal{M}_{hao}$ ).

	DexYCB		FPHAB		HO-3D		Time
Method	<i>AUC</i>	<i>AP</i>	<i>AUC</i>	<i>AP</i>	<i>AUC</i>	<i>AP</i>	<i>ms</i>
w/o hand keypoints	76.2	77.1	76.8	79.3	75.2	77.6	38
w/o $\mathcal{M}_{fus}$	70.5	71.3	71.4	72.1	68.4	70.2	39
w/o $\mathcal{M}_{hao}$	77.8	79.2	77.5	79.7	76.3	77.8	40
Full	<b>80.3</b>	<b>81.2</b>	<b>82.1</b>	<b>83.4</b>	<b>81.5</b>	<b>82.6</b>	40

**Table 5**

Ablation study with Iterative Refinement.

	DexYCB		FPHAB		HO-3D		Time
Method	<i>AUC</i>	<i>AP</i>	<i>AUC</i>	<i>AP</i>	<i>AUC</i>	<i>AP</i>	<i>ms</i>
w/o hand keypoints	84.2	85.1	86.1	86.5	85.2	85.0	190
w/o $\mathcal{M}_{fus}$	75.5	75.6	76.3	76.8	75.4	75.7	268
w/o $\mathcal{M}_{hao}$	84.5	85.6	85.6	86.3	85.8	85.3	192
Full	<b>86.7</b>	<b>87.2</b>	<b>88.0</b>	<b>88.5</b>	<b>88.3</b>	<b>87.7</b>	200

for accurate pose predictions. Removing the hand-aware object pose estimation module ( $\mathcal{M}_{hao}$ ) caused a moderate decrease in performance. The AUC dropped to 77.8, 77.5, and 76.3 on DexYCB, FPHAB, and HO-3D, respectively. The self-attention mechanism that learns the relationship between the hand and the object significantly contributes to the overall accuracy of pose estimation, even though it is not as critical as the vote-based fusion module. The full model, incorporating all components, achieved the highest performance across all datasets and metrics, with an AUC of 80.3, 82.1, and 81.5 on DexYCB, FPHAB, and HO-3D, respectively. This confirms the effectiveness of our proposed method and the synergistic benefits of combining hand keypoint voting, vote-based fusion with channel attention, and hand-aware object pose estimation. The computational time for the full model is 40 ms without iterative refinement and 200 ms with iterative refinement, which, despite a slight increase compared to some ablated versions, remains within a reasonable range. These results justify the additional computational cost, making the full model the most robust and accurate for hand-held object pose estimation.

## 5. Conclusion

In this paper, we have introduced a novel deep neural network designed for the 6D pose estimation of hand-held objects using RGB-D images. Our approach tackles the significant challenges posed by occlusions from the hand and the complexities in effectively fusing RGB and depth data. Our network leverages a voting scheme where both 2D and 3D keypoints cast votes for the object’s pose, significantly enhancing the estimation in scenarios with occluded objects. Additionally, we model the interaction between the hand and the object through a self-attention mechanism, which captures complex spatial relationships and improves the robustness of the pose estimation process. Extensive experiments on three public datasets demonstrate that our method outperforms existing approaches in terms of accuracy and robustness. Our vote-based RGBD fusion framework offers a promising solution for hand-held object pose estimation, particularly in challenging scenarios involving occlusions and complex hand-object interactions. Future work will focus on further improving the efficiency of the model and exploring its application to a broader range of objects and environments.

## References

- Andrychowicz, O.M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., et al., 2020. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research* 39, 3–20.
- Anzai, T., Takahashi, K., 2020. Deep gated multi-modal learning: In-hand object pose changes estimation using tactile and image data, in: *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, IEEE. pp. 9361–9368.
- Billings, G., Johnson-Roberson, M., 2019. Silhonet: An rgb method for 6d object pose estimation. *IEEE Robotics and Automation Letters* 4, 3727–3734.
- Bréquier, R., Devernay, F., Leyrit, L., Crowley, J.L., 2017. Symmetry aware evaluation of 3d object detection and pose estimation in scenes of many parts in bulk, in: *Int. Conf. Comput. Vis. Worksh.*, pp. 2209–2218.
- Castro, P., Kim, T.K., 2023. Crt-6d: Fast 6d object pose estimation with cascaded refinement transformers, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 5746–5755.
- Chao, Y.W., Yang, W., Xiang, Y., Molchanov, P., Handa, A., Tremblay, J., Narang, Y.S., Van Wyk, K., Iqbal, U., Birchfield, S., et al., 2021. Dexycb: A benchmark for capturing hand grasping of objects, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 9044–9053.
- Di, Y., Zhang, R., Lou, Z., Manhardt, F., Ji, X., Navab, N., Tombari, F., 2022. Gpv-pose: Category-level object pose estimation via geometry-guided point-wise voting, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 6781–6791.
- Doosti, B., Naha, S., Mirbagheri, M., Crandall, D.J., 2020. Hope-net: A graph-based model for hand-object pose estimation, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 6608–6617.
- Gao, G., Lauri, M., Wang, Y., Hu, X., Zhang, J., Frintrop, S., 2020. 6d object pose regression via supervised learning on point clouds, in: *Proc. IEEE Int. Conf. Robot. Automat.*, IEEE. pp. 3643–3649.
- Garcia-Hernando, G., Yuan, S., Baek, S., Kim, T.K., 2018. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 409–419.
- Girshick, R., 2015. Fast r-cnn, in: *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 1440–1448.
- Guo, J., Xing, X., Quan, W., Yan, D.M., Gu, Q., Liu, Y., Zhang, X., 2021. Efficient center voting for object detection and 6d pose estimation in 3d point cloud. *IEEE Transactions on Image Processing* 30, 5072–5084.
- Hampali, S., Rad, M., Oberweger, M., Lepetit, V., 2020. Honnotate: A method for 3d annotation of hand and object poses, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 3196–3206.
- Handa, A., Van Wyk, K., Yang, W., Liang, J., Chao, Y.W., Wan, Q., Birchfield, S., Ratliff, N., Fox, D., 2020. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system, in: *Proc. IEEE Int. Conf. Robot. Automat.*, IEEE. pp. 9164–9170.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 770–778.
- He, Y., Huang, H., Fan, H., Chen, Q., Sun, J., 2021. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 3003–3013.
- He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J., 2020. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 11632–11641.
- Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N., 2012. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes, in: *Asian conference on computer vision*, Springer. pp. 548–562.
- Hong, Z.W., Hung, Y.Y., Chen, C.S., 2024. Rdpn6d: Residual-based dense point-wise network for 6dof object pose estimation based on rgb-d images, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 5251–5260.
- Hough, P.V., 1959. Machine analysis of bubble chamber pictures, in: *Proc. of the International Conference on High Energy Accelerators and Instrumentation*, Sept. 1959, pp. 554–556.
- Lin, Y., Su, Y., Nathan, P., Inuganti, S., Di, Y., Sundermeyer, M., Manhardt, F., Stricker, D., Rambach, J., Zhang, Y., 2024. Hipose: Hierarchical binary surface encoding and correspondence pruning for rgb-d 6dof object pose estimation, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 10148–10158.
- Lin, Z., Ding, C., Yao, H., Kuang, Z., Huang, S., 2023. Harmonious feature learning for interactive hand-object pose estimation, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 12989–12998.
- Muhanna, M.A., 2015. Virtual reality and the cave: Taxonomy, interaction challenges and research directions. *Journal of King Saud University-Computer and Information Sciences* 27, 344–361.
- Okafor, E., Oyediji, M., Alfarraj, M., 2024. Deep reinforcement learning with light-weight vision model for sequential robotic object sorting. *Journal of King Saud University-Computer and Information Sciences* 36, 101896.
- Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H., 2019. Pvnnet: Pixel-wise voting network for 6dof pose estimation, in: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 4561–4570.
- Pfanne, M., Chalon, M., Stulp, F., Albu-Schäffer, A., 2018. Fusing joint measurements and visual features for in-hand object pose estimation. *IEEE Robotics and Automation Letters* 3, 3497–3504.
- Qi, C.R., Litany, O., He, K., Guibas, L.J., 2019. Deep hough voting for 3d object detection in point clouds, in: *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 9277–9286.
- Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: *Advances in neural information processing systems*, pp. 5099–5108.
- Qi, H., Zhao, C., Salzmann, M., Mathis, A., 2024. Hoisd: Constraining 3d hand-object pose estimation with global signed distance fields. *arXiv preprint arXiv:2402.17062*.
- Romero, J., Tzionas, D., Black, M.J., 2022. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*.
- Sharma, V.K., Mir, R.N., 2022. Saliency guided faster-rcnn (sgfr-rcnn) model for object detection and recognition. *Journal of King Saud University-Computer and Information Sciences* 34, 1687–1699.

- Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S., 2019. Densefusion: 6d object pose estimation by iterative dense fusion, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 3343–3352.
- Wang, G., Manhardt, F., Tombari, F., Ji, X., 2021a. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 16611–16621.
- Wang, H., Wang, H., Zhuang, C., 2021b. 6d pose estimation from point cloud using an improved point pair features method, in: 2021 7th International Conference on Control, Automation and Robotics (ICCAR), IEEE. pp. 280–284.
- Wang, R., Mao, W., Li, H., 2023. Interacting hand-object pose estimation via dense mutual attention, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 5735–5745.
- Woo, T., Park, W., Jeong, W., Park, J., 2023. A survey of deep learning methods and datasets for hand pose estimation from hand-object interaction images. *Computers & Graphics* 116, 474–490.
- Wu, C., Chen, L., Wang, S., Yang, H., Jiang, J., 2023. Geometric-aware dense matching network for 6d pose estimation of objects from rgb-d images. *Pattern Recognition* 137, 109293.
- Xie, Q., Lai, Y.K., Wu, J., Wang, Z., Lu, D., Wei, M., Wang, J., 2021. Venet: Voting enhancement network for 3d object detection, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 3712–3721.
- Xie, Q., Lai, Y.K., Wu, J., Wang, Z., Zhang, Y., Xu, K., Wang, J., 2020. Mlcvnet: Multi-level context votenet for 3d object detection, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 10447–10456.
- Zhang, H., Goodfellow, I., Metaxas, D., Odena, A., 2019. Self-attention generative adversarial networks, in: International conference on machine learning, PMLR. pp. 7354–7363.