KeyPose: Category-Level 6D Object Pose Estimation with Self-Adaptive Keypoints

Sheng Yu[1], **Di-Hua Zhai**[1*], **Yuanqing Xia**[1,2]

[1]School of Automation, Beijing Institute of Technology, Beijing, China
[2]Zhongyuan University of Technology, Zhengzhou, Henan, China
yusheng@bit.edu.cn, zhaidih@bit.edu.cn, xia_yuanqing@bit.edu.cn.

Abstract

Category-level object pose estimation is an important task in computer vision. Some prior methods based on assumptions often struggle with drastic changes in object appearance. To address this challenge, we propose a new method for object pose estimation based on object-adaptive keypoints. In this paper, we first introduce a transformer-based keypoint prediction method for adaptive forecasting of point cloud keypoints. This method calculates the similarity between keypoint features and point cloud features, allowing keypoints to represent object geometry more effectively. Furthermore, to enhance the geometric feature construction of keypoints, we propose a graph-based keypoint feature aggregation method, which considers both the structural relationships between keypoints and the point cloud, strengthening the network's understanding of geometric structures. At this stage, keypoints remain at the geometric spatial level of the object and have not been predicted in NOCS. To improve the accuracy of keypoint prediction in NOCS, we design a NOCS voxelization method that divides NOCS into multiple voxels and accurately predicts NOCS keypoints within these voxels. Experimental results on multiple benchmark datasets demonstrate that our proposed KeyPose method outperforms all existing methods, achieving over 20% improvement in pose accuracy on some critical datasets.

1 Introduction

Object pose estimation is an important task in the field of computer vision and is widely applied in robotics manipulation (Wang et al. 2019a; Lin et al. 2022a), augmented reality (Tjaden, Schwanecke, and Schomer 2017), autonomous driving (Manhardt, Kehl, and Gaidon 2019), and other fields. Previous pose estimation methods have often focused on instance-level object pose estimation, such as (Xiang et al. 2018; Kehl et al. 2017; Peng et al. 2019; He et al. 2020), but these methods rely on the object's 3D model, making them difficult to apply to pose estimation tasks involving real-world unknown objects. To address this issue, researchers have proposed category-level object pose estimation methods, such as (Wang et al. 2019b; Tian, Ang, and Lee 2020; Chen et al. 2021; Chen and Dou 2021; Lin et al. 2021; Chen et al. 2020a; Di et al. 2022; Lin et al. 2022a).

*Corresponding author.

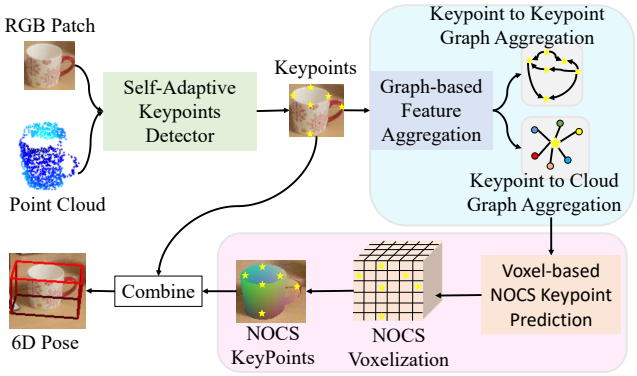

Figure 1: KeyPose takes the RGB and the point cloud as the input, and detects the keypoints with a self-adaptive keypoints detector.

Compared to instance-level object pose estimation methods, category-level object pose estimation methods do not rely on the object's 3D model. They can effectively predict the poses of all unknown objects within the same category, demonstrating strong practical applicability and generalization performance. But due to variations in shape among different instances within the same category, category-level object pose estimation faces significant challenges.

Currently, most methods achieve object pose estimation by establishing correspondences between the object's model in Normalized Object Coordinate Space (NOCS) and the actual object point cloud. Some methods, such as (Tian, Ang, and Lee 2020; Chen and Dou 2021; Lin et al. 2022a), based on prior point clouds for pose estimation usually deform the prior point cloud and establish dense correspondences between the deformed prior point cloud and the actual point cloud, thereby predicting the object's pose. However, these methods of predicting poses through deformation and subsequent correspondence often encounter local matching errors. Additionally, studies like (Liu et al. 2023) have demonstrated that prior point clouds are unnecessary for category-level object pose estimation. Therefore, we need a direct and effective approach to replace prior point clouds for accurately representing the 3D structure of objects.

We notice that each object category contains distinct rep-

resentative regions, like a camera lens or a mug handle. Representing these regions with keypoints enables capturing the overall geometric structure of objects. However, category-level object pose estimation faces challenges due to the absence of 3D object models, complicating keypoint localization. Recent attempts at predicting keypoints for category-level object pose, such as few-shot methods (e.g., (Lin et al. 2024)), focus solely on local object features without considering global features. Moreover, these methods overlook differences between object and keypoint features, relying solely on direct cosine similarity for measurement.

To address these issues, we introduce KeyPose, a novel category-level object pose estimation method based on self-adaptive keypoints, as shown in Figure 1. Initially, we integrate local and global features, and propose a self-adaptive keypoint prediction module. Using these features, we employ transformers to predict initial keypoint features. To ensure accurate representation of object fused features by initial keypoint features, we develop a keypoint feature prediction method. This involves random sampling of fused features based on the number of point clouds and employing transformers to compute similarity between downsampled and initial keypoint features, achieving a comprehensive representation. Furthermore, to fully leverage keypoints and key features, we propose a graph-based feature aggregation algorithm. We construct a graph structure between keypoints and point clouds using the nearest points around the keypoints. This graph structure between keypoints and keypoints enhances the network's understanding of keypoints by extracting geometric and graph features.

Currently, we've achieved keypoint prediction in geometric space but haven't extended this to NOCS keypoints. NOCS keypoints are vital for accurately predicting object poses, but direct prediction from geometric keypoints can lead to positional inaccuracies. To resolve this, we propose a NOCS keypoint prediction method based on spatial voxel partitioning. This approach partitions the NOCS space into voxels, enabling precise keypoint prediction within each voxel and markedly improving NOCS keypoint accuracy.

Finally, we conduct performance tests on multiple benchmark datasets. Experimental results indicate that our proposed KeyPose method achieves more accurate pose estimation across various datasets. In some datasets, it even demonstrates an improvement in accuracy of over 20%. We have achieved excellent results in predicting poses.

In summary, the main contributions of this paper are summarized as follows:

- We propose a novel transformer-based self-adaptive keypoint detection algorithm, achieving adaptive prediction of keypoints for different categories and objects, enhancing pose estimation accuracy.
- We propose a graph-based feature aggregation algorithm to extract and aggregate graph features of keypoints and point clouds.
- We propose a voxel-based keypoint prediction method that accurately predicts keypoints of unknown objects in NOCS.

2 Related Works

2.1 Instance-level 6D Object Pose Estimation

The methods of instance-level 6D object pose estimation can be mainly divided into three categories: template matching-based, regression-based, and corresponding-based methods. For the template matching methods, such as (Hinterstoisser et al. 2011; Oberweger, Rad, and Lepetit 2018; Rad, Oberweger, and Lepetit 2018; Wu and Greenspan 2024), these methods try to align the 3D model of the object to the RGB-D image based on the extracted features, which may consume much time in the matching process. To solve this problem, some researchers propose regression-based methods, such as (Xiang et al. 2018; Kehl et al. 2017). These methods take RGB images as input, and directly output the object 6D pose. However, the accuracy of these methods is relatively low. To solve this problem, some authors propose correspondences-based methods, such as (Peng et al. 2019; Rad and Lepetit 2017; He et al. 2020). These methods try to establish 2D-3D correspondences or 3D-3D correspondences, where the 6D pose is calculated by solving Perspective-n-Point (PnP) and SVD problems. All these methods are trained in a fully-supervised manner, and the 3D CAD models of objects are required.

2.2 Category-level 6D Object Pose Estimation

Although instance-level 6D object pose estimation has got great progress, it relies heavily on the 3D CAD models of objects, which means it cannot handle the unseen object. In contrast, category-level 6D object pose estimation methods aim to predict the pose of unseen objects.

To further enhance pose estimation accuracy, some methods are attempting to introduce prior point clouds. In (Tian, Ang, and Lee 2020), Tian et al. propose to introduce the prior point cloud to improve the estimation accuracy. In (Chen and Dou 2021), the authors propose SGPA, which uses a vision transformer module to fuse the features from the prior point cloud, the scene point cloud, and the RGB image. CatFormer (Yu, Zhai, and Xia 2024) introduces prior point clouds to enhance pose prediction accuracy and designs a method for point cloud deformation and optimization to iteratively deform and optimize the prior point clouds.

Furthermore, recent research has introduced prior-free methods. In (Lin et al. 2021), Lin et al. propose DualPoseNet, which utilizes a dual channel structure to perform feature extraction and fusion, and a novel pose refinement method is also proposed by means of pose consistency. In (Chen et al. 2021), Chen et al. propose FS-Net, which tries to represent rotation by using decoupled vectors. VI-Net (Lin et al. 2023) attains high precision in object pose estimation by separating rotation into viewpoint and in-plane rotations.

3 Method

3.1 Pipeline Overview

The pipeline of the training process of KeyPose is shown in Figure 2. We take the RGB image and depth image as inputs and utilize an instance segmentation network, such as Mask-RCNN (He et al. 2017), to segment the object mask from

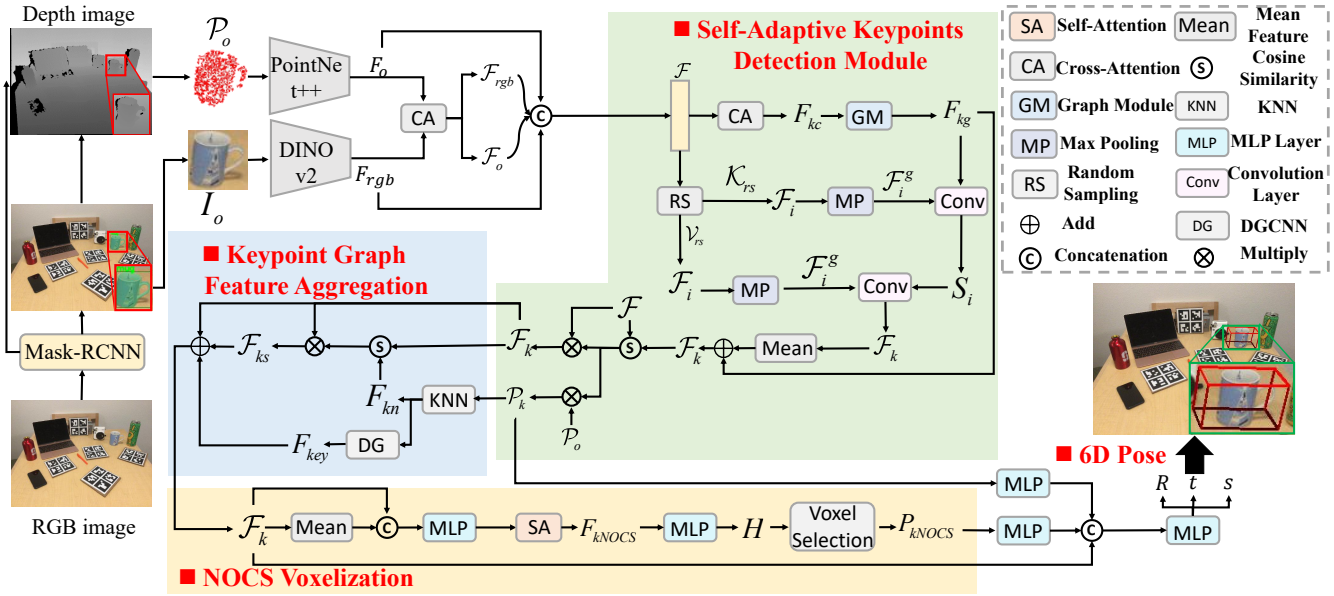

Figure 2: The pipeline of the KeyPose. We initially employ Mask-RCNN to predict the mask and category of the target object. Then KeyPose takes the RGB patch and the point cloud as the input. We first employ a self-adaptive keypoints module to detect the adaptive keypoints. Then, we use a keypoint graph feature aggregation module to establish the graph feature of the keypoints and point cloud. Finally, we voxelize the NOCS to predict the keypoints in the NOCS based on keypoint features. Based on the predicted keypoints in geometric space and NOCS, we predict the 6D pose of the target object.

the image. Subsequently, we crop the RGB image and depth image based on this instance mask, resulting in the RGB patch and depth patch of the target object. By leveraging the camera's intrinsic matrix, we can generate the point cloud of the target object. We define $\mathcal{P}_o \in \mathbb{R}^{N_o \times 3}$ as the point cloud of the target object, which contains N_o points, $I_o \in \mathbb{R}^{H \times W \times 3}$ is the RGB image of the object. Based on these inputs, we will use KeyPose to estimate the 6D pose of the unseen object.

3.2 Self-Adaptive Keypoints Detection Module

Following previous methods (Chen et al. 2024) of feature processing for RGB images and point clouds, in this paper, we employ DINOv2 (Oquab et al. 2023) as the backbone for RGB images and PointNet++ (Qi et al. 2017) as the backbone for point clouds to obtain the RGB features $F_{rgb} \in \mathbb{R}^{N_o \times d}$ and point cloud features $F_o \in \mathbb{R}^{N_o \times d}$ of the object. We treat F_{rgb} and F_o as the local features, and calculate the global features with transformer.

First, we calculate the global features of F_{rgb} and F_o with mean pooling, and get the global-wise features $\mathcal{F}_{rgb} \in \mathbb{R}^{1 \times d}$ and $\mathcal{F}_o \in \mathbb{R}^{1 \times d}$. Then we generate query q_*, key k_*, and value v_* based on the extracted features, where $* \in \{rgb, o\}$.

In order to fuse the feature from the RGB image and point cloud, we use the cross-attention module to fuse them, which can be calculated by $\mathcal{F}_i = \text{CA}_{j \to i}(q_i, k_j, v_j)$, where $\text{CA}_{j \to i}, i, j \in \{rgb, o\}, i \neq j$ indicates the cross-attention of point cloud to RGB and RGB to point cloud respectively.

The calculation process is indicated as

$$\text{CA} = softmax\left(\frac{q_i \times k_j^T}{\sqrt{d_k}}\right) v_j + q_i \tag{1}$$

Then, we repeatedly replicate the global feature $\mathcal{F}_{rgb}$ and $\mathcal{F}_o$ with N_o times, and concatenate it with the local features along the feature dimension to obtain fused features.

$$\mathcal{F} = ©(\mathcal{F}_{rgb}, F_{rgb}, \mathcal{F}_o, F_o) \tag{2}$$

where $\mathcal{F} \in \mathbb{R}^{N_o \times 4d}$, © indicates concatenation operations.

Based on $\mathcal{F}$, we calculate keypoint features and further predict keypoints. We first randomly initialize a learnable keypoint feature $F_k \in \mathbb{R}^{N_k \times 4d}$, where N_k indicates the number of keypoints, $N_k < N_o$.

Then, we initialize F_k based on $\mathcal{F}$ using a transformer. Our goal is to allow F_k to represent key features from $\mathcal{F}$. Here, F_k serves as the query, while $\mathcal{F}$ serves as the key and value of the transfomer. We compute the initial keypoint features using cross-attention $F_{kc} = \text{CA}(F_k, \mathcal{F}, \mathcal{F})$.

Further, to achieve adaptive adjustment of initial keypoint features and extraction of geometric information, we utilize the transformer-based graph module (GM) proposed in (Yu, Zhai, and Xia 2024) for self-attention computation and extraction of graph information, $F_{kg} = \text{GM}(F_{kc})$.

The keypoints we predict are essential representative points in the point cloud. We select these points to effectively capture the object's features across diverse conditions and withstand outliers. To achieve this, we develop a transformer-based method for predicting keypoint features.

Firstly, keypoint features can characterize object features at any given moment and mitigate the impact of outliers. Therefore, we randomly sample the fused feature $\mathcal{F}$ along the dimension of point quantity to obtain a random set of point features. We use this set as the key and value for the transformer, $\mathcal{K}_{rs} = \mathcal{V}_{rs} = \{\mathcal{F}_1, \cdots, \mathcal{F}_M\}$, where M represents the number of samples taken, $\mathcal{F}_i \in \mathbb{R}^{N_k \times 4d}, i \in \{1, \cdots, M\}$.

Then for each $\mathcal{F}_i \in \mathcal{K}_{rs}$, we calculate the similarities between the $\mathcal{F}_i$ and F_{kg}. We apply a global max pooling operation on $\mathcal{F}_i$ to compute its global feature, which encapsulates crucial information from the sampled point features. Therefore, the keypoints we need to select are those that maintain a high similarity with the sampled point features regardless of the sampling method. Hence, we have devised a convolution-based similarity calculation method. By performing global pooling, we obtain the global feature $\mathcal{F}_i^g \in \mathbb{R}^{N_k \times 1}$ of $\mathcal{F}_i$. Subsequently, using $\mathcal{F}_i^g$ as a convolutional kernel, we compute the convolutional values on F_{kg}

$$S_i = \text{Conv}(F_{kg}, \text{k} = \mathcal{F}_i^g) \tag{3}$$

where $S_i \in \mathbb{R}^{N_k \times 4d}$ indicates the similarity map, k indicates the kernel of the convolution layer.

Afterwards, we obtain M similarity maps like this and concatenate these maps to form the complete set of similarity maps $S_M \in \mathbb{R}^{M \times N_k \times 4d}$. To enhance efficiency during training, we apply exemplar normalization (EN) and spatial normalization (SN) to each similarity map separately. The final similarity map S_M is then computed as the product of the normalized versions of these maps.

Finally, we further adjust the features $\mathcal{F}_i \in \mathcal{V}_{rs}$ using the similarity maps S_M. Similar to the aforementioned process, we apply global pooling to $\mathcal{F}_i$ to obtain $\mathcal{F}_i^g$, and then use it as a convolutional kernel. Through the similarity sub-maps S_i, we adjust each feature to achieve the desired keypoint features $F_i = \text{Conv}(S_i, \text{k} = \mathcal{F}_i^g)$.

We concatenate the obtained F_i to obtain the keypoint features $\mathcal{F}_k \in \mathbb{R}^{M \times N_k \times 4d}$. To simultaneously consider these features, we compute their average feature and fuse it with the initial keypoint features by addition $\mathcal{F}_k \leftarrow \mathbf{M}(\mathcal{F}_k) + F_{kg}$, where $\mathbf{M}$ indicates the mean calculation.

We also employ cosine similarity at the end to assess the similarity between the keypoint features and the fused features. Based on this similarity $\mathcal{S} \in \mathbb{R}^{N_k \times N_o}$, we make final adjustments and compute the ultimate keypoint features $\mathcal{F}_k$ and keypoints $\mathcal{P}_k$.

$$\mathcal{F}_k = softmax(\mathcal{S}) \times \mathcal{F}, \mathcal{P}_k = softmax(\mathcal{S}) \times \mathcal{P}_o \tag{4}$$

3.3 Keypoint Graph Feature Aggregation

Although we can predict poses using keypoint features, we lack geometric information regarding the relationships between keypoints and object point clouds. Establishing a geometric structure graph among points could enhance the network's comprehension of object geometry. Thus, we introduce a keypoint-oriented graph feature aggregation algorithm to address these issues.

Firstly, we consider the graph structural features between keypoints and the object point cloud. We search for the K nearest points in the point cloud for each keypoint. This allows us to obtain a series of nearest neighbor sets $P_{kn} \in \mathbb{R}^{K \times N_k \times 3}$ and the corresponding nearest neighbor features $F_{kn} \in \mathbb{R}^{K \times N_k \times 4d}$ from the object point cloud. Furthermore, we aim to ensure that the predicted keypoints are highly representative, meaning that each keypoint can characterize the geometric features represented by its surrounding K keypoints. Therefore, we also compute the cosine similarity between the nearest neighbor features F_{kn} and the keypoint features $\mathcal{F}_k$, generating a similarity map $\mathcal{S}_{kn}$ which is used to adjust the keypoint features $\mathcal{F}_{ks} = softmax(\mathcal{S}_{kn}) \times \mathcal{F}_k$

Then, we need to consider the graph features between keypoints. Here, we utilize DGCNN (Wang et al. 2019c) as the feature extraction network, taking the K_{key} nearest keypoints around each keypoint to construct a keypoint graph and extract the graph features F_{key} from the keypoints.

Combining the above features, we can obtain the fused features of the keypoints $\mathcal{F}_k \leftarrow \mathcal{F}_{ks} + F_{key} + \mathcal{F}_k$.

Simultaneously, to better integrate the keypoint features, we compute their feature averages, treating them as global features and concatenate them with $\mathcal{F}_k$,

$$\mathcal{F}_k \leftarrow \copyright(\mathbf{M}(\mathcal{F}_k), \mathcal{F}_k) \tag{5}$$

3.4 Pose Estimation

At this moment, we need to estimate the object pose based on keypoint features. The keypoints we predict are on the object point cloud level, representing the geometric space of the object. Additionally, we need to predict the keypoint positions of the object in NOCS. While direct prediction using features is possible, the disparity between these two spaces might introduce positional deviations in the predicted keypoints. To address this issue, we propose a NOCS keypoint prediction method based on 3D voxel partitioning. Given that NOCS is confined within a $1 \times 1 \times 1$ space, we initially partition this space into multiple cubes with side length a, forming the entire space using $\mathcal{A} = \frac{1}{a} \times \frac{1}{a}$ such cubes. Compared to the positional bias introduced by direct prediction, here we mitigate errors by predicting which voxel contains the keypoints, using the center of the voxel as the location for NOCS keypoints.

Firstly, based on the keypoint features, we directly predict the initial NOCS (Normalized Object Coordinate Space) keypoint features using an MLP layer. Then, we use a self-attention layer to adjust the initial NOCS keypoint features, $F_{kNOCS} = \text{SA}(\text{MLP}(\mathcal{F}_k))$.

Furthermore, based on F_{kNOCS}, we predict a keypoint heatmap $H \in \mathbb{R}^{3 \times \mathcal{A} \times N_k}$ through an MLP layer. The keypoint heatmap H scores the presence of keypoints across three dimensions of the point cloud, selecting the center of the voxel that scores highly in all three dimensions as the location of the keypoints in NOCS. This process allows us to determine the positions of all keypoints P_{kNOCS} in NOCS.

Finally, based on geometric keypoints, NOCS keypoints, and keypoint features, we predict the pose of the object

$$R, t, s = \text{MLP}(\copyright(\text{MLP}(P_{kNOCS}), \text{MLP}(\mathcal{P}_k), \mathcal{F}_k)) \tag{6}$$

3.5 Loss Function

Firstly, key points of objects must maintain a certain distance from each other, or else they will form small-scale clusters

$$L_d = \sum_{i=1}^{N_k} \sum_{j=1, j\neq i}^{N_k} d(\mathcal{P}_k^{(i)}, \mathcal{P}_k^{(j)}) \tag{7}$$

$$d(\mathcal{P}_k^{(i)}, \mathcal{P}_k^{(j)}) = max\{\alpha - ||\mathcal{P}_k^{(i)} - \mathcal{P}_k^{(j)}||_2, 0\} \tag{8}$$

where α is the distance threshold, $\mathcal{P}_k^{(i)}$ is the i-th keypoint.

Then, we aim for the keypoints to be as close as possible to the surface of the object

$$L_D = \frac{1}{N_k} \sum_{\mathcal{P}_k^{(i)} \in \mathcal{P}_k} \min_{\mathcal{P}_{gt}^{(i)} \in \mathcal{P}_{gt}} ||\mathcal{P}_k^{(i)} - \mathcal{P}_{gt}^{(i)}||_2 \tag{9}$$

where $\mathcal{P}_{gt}$ indicates the ground truth point cloud with outlier removal.

Simultaneously, the NOCS keypoints of the object also need to be constrained to ensure alignment with the actual NOCS keypoint positions

$$L_{kNOCS} = L_1(P_{kNOCS}, \hat{P}_{kNOCS}) \tag{10}$$

where L_1 indicates the SmoothL1 loss, $\hat{P}_{kNOCS}$ indicates the ground truth keypoints in NOCS.

Finally, we also need to impose restrictions on the pose.

$$L_{Pose} = ||R - \hat{R}||_2 + ||t - \hat{t}||_2 + ||s - \hat{s}||_2 \tag{11}$$

where $\hat{R}, \hat{t}, \hat{s}$ indicate the ground truth.

The total loss is

$$L = \lambda_1 L_d + \lambda_2 L_D + \lambda_3 L_{kNOCS} + \lambda_4 L_{Pose} \tag{12}$$

where λ_* indicates the hyperparameter. We set $\lambda_1 = 1, \lambda_2 = 5, \lambda_3 = 1, \lambda_4 = 0.3$.

4 Experiments

4.1 Datasets

CAMERA25 and REAL275 Dataset The benchmark datasets used for category-level object pose estimation are the REAL275 dataset and the CAMERA25 dataset (Wang et al. 2019b). The CAMERA25 dataset comprises 300K images, with 25K images designated for evaluation. It is created by integrating synthetic objects into real-world scenes. In contrast, the REAL275 dataset consists of 4300 real-world training images from 7 scenes and 2750 real-world evaluation images from 6 scenes. Both datasets contains: bottle, bowl, camera, can, laptop, and mug.

HouseCat6D Dataset HouseCat6D (Jung et al. 2024) is a comprehensive real-world dataset that includes 194 high-fidelity 3D models of household items across 10 categories. The dataset features transparent and reflective objects placed in 41 scenes, providing diverse viewpoints, challenging occlusions, and lacking any markers.

4.2 Evaluation Metrics

We evaluate the performance of KeyPose using widely used evaluation metrics (Wang et al. 2019b; Tian, Ang, and Lee 2020; Lin et al. 2021; Di et al. 2022; Lin et al. 2022a). For rotation and translation evaluation, we utilize 3D Intersection-Over-Union (IoU) with thresholds of 0.5, and 0.75. Additionally, we employ $5°2cm$, $5°5cm$, $10°2cm$, and $10°5cm$ to directly assess rotation and translation accuracy. If the errors fall within the thresholds, the predictions are deemed correct. Based on these evaluation metrics, we will use overall mAP to assess the performance of KeyPose compared to other SOTA methods.

4.3 Comparison with State-of-the-Art Methods

Results on CAMERA25 and REAL275 dataset First, we conduct experiments on benchmark datasets for category-level object pose estimation, including CAMERA25 and REAL275 datasets. The experimental results are shown in Table 1. Due to the REAL275 dataset being collected in actual scenes, it is more susceptible to real-world noise, thus better reflecting the robustness of pose estimation methods against interference. The CAMERA25 dataset, collected in simulated environments, is unaffected by environmental noise. Therefore, many methods currently only undergo performance testing on the REAL275 dataset. Here, we only compare methods that have been experimented with on both CAMERA25 and REAL275 datasets.

KeyPose demonstrates superior performance across different datasets. On the CAMERA25 dataset, which offers stable depth information unaffected by environmental factors, KeyPose outperforms existing methods by approximately 2%. Moreover, on the REAL275 dataset, KeyPose shows significant performance gains compared to state-of-the-art methods like VI-Net and AG-Pose. Specifically, in metrics like $5°5cm$, KeyPose achieves around a 6% improvement. These advancements are attributed to our adaptive keypoint prediction algorithm, which effectively handles object shape variations across categories, thereby enhancing pose estimation accuracy.

In Figure 3, we present a detailed qualitative comparison of KeyPose with VI-Net and AG-Pose on the REAL275 dataset. Predictions are marked with red bounding boxes, while ground truth is indicated by green bounding boxes. Objects within the yellow dashed box illustrate discrepancies in pose estimation between VI-Net and AG-Pose compared to KeyPose. Visual results demonstrate that KeyPose achieves superior accuracy in pose estimation.

Results on HouseCat6D dataset To further validate the performance of our proposed method in pose estimation for other categories, we conduct pose estimation experiments on the HouseCat6D (Jung et al. 2024) dataset. Compared to the standard CAMERA25 and REAL275 datasets, the HouseCat6D dataset includes a wider range of object categories. The experimental results are shown in the Table 2. We follow the pose accuracy evaluation method of the HouseCat6D dataset and report the accuracy of all objects at 3D IoU thresholds of 25% and 50%.

Method	Prior	CAMERA25						REAL275						FPS
		IoU_{50}	IoU_{75}	5°2cm	5°5cm	10°2cm	10°5cm	IoU_{50}	IoU_{75}	5°2cm	5°5cm	10°2cm	10°5cm	
SPD(Tian, Ang, and Lee 2020)	✓	93.2	83.1	54.3	59.0	73.3	81.5	77.3	53.2	19.2	21.4	43.2	54.1	5.9
SGPA(Chen and Dou 2021)	✓	93.2	88.1	70.7	74.5	82.7	88.4	80.1	61.9	35.9	39.6	61.3	70.7	5.0
CR-Net(Wang, Chen, and Dou 2021)	✓	93.8	88.0	72.0	76.4	81.0	87.7	79.3	55.9	27.8	34.3	47.2	60.8	-
RBP-Pose (Zhang et al. 2022)	✓	93.1	89.0	73.5	79.6	82.1	89.5	-	67.8	38.2	48.1	63.1	79.2	24.3
SAR-Net(Lin et al. 2022a)	✓	86.8	79.0	66.7	70.9	75.3	80.3	79.3	62.4	31.6	42.3	50.3	68.3	-
CatFormer(Yu, Zhai, and Xia 2024)	✓	93.5	89.9	74.9	79.8	85.3	90.2	83.1	73.8	47.7	53.7	69.0	79.5	8.0
NOCS(Wang et al. 2019b)		83.9	69.5	32.3	40.9	48.2	64.6	78.0	30.1	7.2	9.5	13.8	25.2	4.8
DualPoseNet(Lin et al. 2021)		92.4	86.4	64.7	70.7	77.2	84.7	79.8	62.2	29.3	35.9	50.0	66.8	2.3
GPV-Pose(Di et al. 2022)		93.4	88.3	72.1	79.1	-	89.0	83.0	64.4	32.0	42.9	-	73.3	22.5
HS-Pose(Zheng et al. 2023)		93.3	89.4	73.3	80.5	80.4	89.4	82.1	74.7	46.5	55.2	68.6	82.7	16.7
VI-Net(Lin et al. 2023)			79.1	74.1	81.4	79.3	87.3	-	48.3	50.0	57.6	70.8	82.1	-
AG-Pose(Lin et al. 2024)		93.8	91.3	77.8	82.8	85.5	91.6	83.7	79.5	54.7	61.7	74.7	83.1	26.8
ours		**94.4**	**92.6**	**79.8**	**83.6**	**87.1**	**92.3**	**84.2**	**80.0**	**57.7**	**66.0**	**78.8**	**88.0**	25.4

Table 1: Comparison with state-of-the-art methods on CAMERA25 dataset and REAL275 dataset.

Approach	IoU_{25}	IoU_{50}	Bottle	Box	Can	Cup	Remote	Teapot	Cutlery	Glass	Tube	Shoe
NOCS	50.0	21.2	41.9 / 5.0	43.3 / 6.5	81.9 / 62.4	68.8 / 2.0	81.8 / 59.8	24.3 / 0.1	14.7 / 6.0	95.4 / 49.6	21.0 / 4.6	26.4 / 16.5
FS-Net	74.9	48.0	65.3 / 45.0	31.7 / 1.2	98.3 / 73.8	96.4 / 68.1	65.6 / 46.8	69.9 / 59.8	71.0 / 51.6	99.4 / 32.4	79.7 / 46.0	71.4 / 55.4
GPV-Pose	74.9	50.7	66.8 / 45.6	31.4 / 1.1	98.6 / 75.2	96.7 / 69.0	65.7 / 46.9	75.4 / 61.6	70.9 / 52.0	99.6 / 62.7	76.9 / 42.4	67.4 / 50.2
VI-Net	80.7	56.4	90.6 / 79.6	44.8 / 12.7	99.0 / 67.0	96.7 / 72.1	54.9 / 17.1	52.6 / 47.3	89.2 / 76.4	99.1 / 93.7	**94.9** / 36.0	85.2 / 62.4
AG-Pose	81.8	62.5	82.3 / 62.8	57.2 / 7.7	97.1 / 83.6	97.9 / 79.6	87.0 / 66.2	63.4 / 60.9	77.2 / 62.0	100.0 / 99.4	83.4 / 53.4	72.0 / 50.0
SecondPose	83.7	66.1	94.5 / 79.8	54.5 / 23.7	98.5 / 93.2	99.8 / 82.9	53.6 / 35.4	81.0 / 71.0	93.5 / 74.4	99.3 / 92.5	75.6 / 35.6	86.9 / 73.0
ours	**87.2**	**75.4**	**94.9 / 90.4**	**58.5 / 17.6**	**98.7 / 93.8**	**100.0 / 99.8**	**62.2 / 45.3**	**84.0 / 76.3**	**95.9 / 94.4**	**100.0 / 99.5**	81.2 / **41.2**	**96.1 / 95.6**

Table 2: Quantitative benchmark comparisons on HouseCat6D dataset.

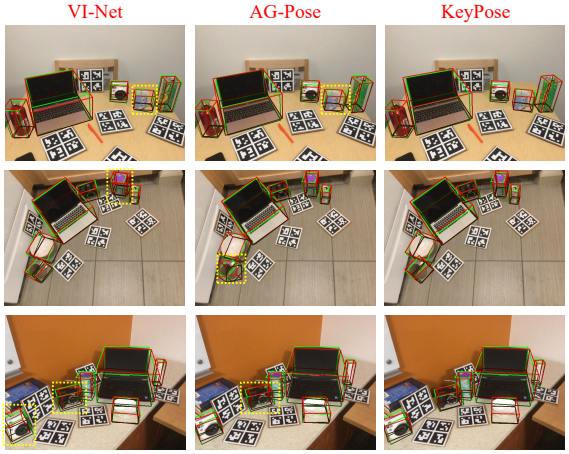

Figure 3: Qualitative results of VI-Net, AG-Pose, and KeyPose on REAL275 dataset.

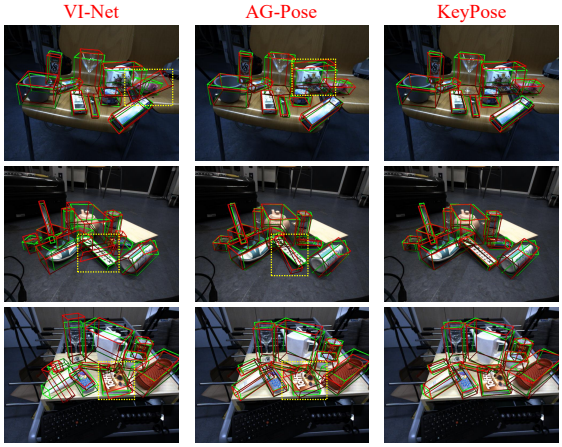

Figure 4: Qualitative results of VI-Net, AG-Pose, and KeyPose on HouseCat6D dataset.

From the results in the Table 2, it can be seen that compared to some previous state-of-the-art methods, KeyPose is able to more accurately perform object pose estimation. It has achieved performance improvements in most categories of pose prediction. For relatively strict pose estimation metrics such as IoU_{50}, the average accuracy has improved by approximately 10%, achieving more precise results.

Experimental results indicate that KeyPose achieves nearly 100% accuracy in transparent object, primarily glass, pose estimation under specified metrics. For items like cutlery and shoes, we achieve a notable 20% enhancement in pose accuracy at the IoU_{50} metric, showcasing outstanding performance. To highlight KeyPose's advantages on the HouseCat6D dataset, we showcase visual comparisons with VI-Net in Figure 4. Ground truth is depicted with green 3D bounding boxes, while red 3D bounding boxes show predicted poses. These visualizations demonstrate that KeyPose excels in accurately estimating object poses, particularly in cluttered and multi-category scenes, showcasing its robust performance.

Group	Module			Keypoints	Random Sample Groups	Loss Terms			REAL275					
	KD	GA	Vox			L_d	L_D	L_{kNOCS}	IoU_{50}	IoU_{75}	$5°2cm$	$5°5cm$	$10°2cm$	$10°5cm$
(A)	✓			96	32	✓	✓	✓	81.7	71.3	37.8	45.8	59.4	70.3
		✓		96	32	✓	✓	✓	82.1	70.4	38.7	45.2	60.1	71.2
			✓	96	32	✓	✓	✓	82.4	72.1	41.5	47.7	63.2	74.5
	✓	✓		96	32	✓	✓	✓	83.1	73.2	44.1	50.7	66.8	75.9
		✓	✓	96	32	✓	✓	✓	83.8	77.5	52.3	59.8	73.1	83.6
	✓		✓	96	32	✓	✓	✓	83.4	76.8	48.3	56.1	72.4	82.5
(B)	✓	✓	✓	16	32	✓	✓	✓	82.5	75.8	49.2	57.8	69.4	79.8
	✓	✓	✓	32	32	✓	✓	✓	83.1	77.5	51.3	59.1	74.0	82.3
	✓	✓	✓	64	32	✓	✓	✓	83.6	78.9	54.4	62.8	76.5	85.9
	✓	✓	✓	96	32	✓	✓	✓	**84.2**	**80.0**	**57.7**	**66.0**	**78.8**	**88.0**
	✓	✓	✓	128	32	✓	✓	✓	83.3	79.5	56.4	64.3	77.1	87.5
(C)	✓	✓	✓	96	8	✓	✓	✓	82.0	71.4	29.2	32.3	52.7	65.8
	✓	✓	✓	96	16	✓	✓	✓	82.5	75.6	44.2	54.7	69.2	77.3
	✓	✓	✓	96	32	✓	✓	✓	**84.2**	**80.0**	**57.7**	**66.0**	**78.8**	**88.0**
	✓	✓	✓	96	64	✓	✓	✓	84.2	79.5	56.4	65.3	77.2	87.3
	✓	✓	✓	96	96	✓	✓	✓	84.2	79.4	57.2	65.5	77.7	87.2
(D)	✓	✓	✓	96	32	✓			83.8	75.9	45.8	55.4	67.9	80.8
	✓	✓	✓	96	32		✓		79.6	60.8	32.8	39.4	53.9	68.4
	✓	✓	✓	96	32			✓	78.6	61.5	30.3	38.2	54.8	68.7
	✓	✓	✓	96	32	✓	✓		84.0	78.5	52.5	61.2	69.0	80.8
	✓	✓	✓	96	32		✓	✓	78.8	62.1	32.5	39.3	56.4	69.6
	✓	✓	✓	96	32	✓		✓	83.5	76.1	48.2	55.3	75.7	83.6

Table 3: Ablation studies on different configurations of network on REAL275. KD indicates the keypoints detection module, GA indicates the graph-based feature aggregation module, and Vox indicates the NOCS voxelization.

4.4 Ablation Studies

To validate the effectiveness of the proposed module, we first conduct a series of ablation experiments on the module. The experimental results are shown in Table 3(A). Due to the necessity of pose prediction relying on key points, when the key point detection module is not used, we directly employ the initial key point features F_k for predicting the key point features and calculate the cosine similarity.

From the experimental results, we can see that removing any module leads to a significant decrease in the network's performance. Particularly, when the NOCS voxelization module is removed, the network's ability to locate keypoints in the NOCS space decreases, resulting in greater pose estimation errors.

Next, we conduct experiments on the number of object keypoints. The selection of keypoints must be appropriate, too few keypoints lack representativeness, while too many can introduce redundancy. Therefore, we determine the optimal number of keypoints through experiments. The experimental results are shown in Table 3(B). From the experimental results, it can be seen that when we choose 96 keypoints, the overall performance of the network reaches its peak.

We need to determine the number of groups for random sampling. If the number of sampling groups is too small, it may not effectively represent the object's characteristics. Conversely, if there are too many groups, it will increase additional computational overhead and time costs. Therefore, we explore the number of groups for random sampling, and the experimental results are shown in Table 3(C). According to the experimental results, we can see that the performance reaches its peak when we choose 32 groups for sampling.

Finally, we conduct ablation experiments on the loss functions. Since the pose loss function is absolutely necessary, we only perform ablation experiments on the other three loss functions, and the experimental results are shown in Table 3(D). According to the experimental results, we can see that when we remove L_d, the network's performance significantly declines. This is because L_d is primarily used to prevent local clustering of keypoints, allowing them to disperse across various crucial areas of the object. Removing it results in the loss of keypoints in critical regions of the object, thus affecting object pose estimation. When we remove L_D, keypoints tend to deviate to some extent from the object's surface. However, because keypoints possess a certain capability for geometric structural representation, the network experiences a decrease in performance after removing L_D. Finally, when we remove L_{kNOCS}, the network can only rely on predicting approximate NOCS keypoints based on actual object keypoints, resulting in a slight performance drop. When we apply all loss functions to KeyPose, the network achieves optimal performance.

5 Conclusion

In this paper, we propose a new category-level object pose estimation method called KeyPose. Firstly, we devise an adaptive keypoint prediction method based on transformers, enabling adaptive prediction of keypoints for different objects. Furthermore, we introduce a graph-based keypoint feature aggregation method to enhance the exploration of keypoint and point cloud features. Lastly, we develop a NOCS keypoint prediction method based on voxelization, achieving more accurate object pose estimation. Multiple experiments demonstrate that our proposed method outperforms all existing state-of-the-art methods.

Acknowledgments

The work was supported by the National Natural Science Foundation of China under Grant 62173035, Grant 61803033 and Grant 61836001, and in part by the Xiaomi Young Scholars from Xiaomi Foundation.

References

Chen, D.; Li, J.; Wang, Z.; and Xu, K. 2020a. Learning canonical shape space for category-level 6D object pose and size estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 11973–11982.

Chen, K.; and Dou, Q. 2021. SGPA: Structure-guided prior adaptation for category-level 6D object pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2773–2782.

Chen, W.; Jia, X.; Chang, H. J.; Duan, J.; and Leonardis, A. 2020b. G2l-net: Global to local network for real-time 6D pose estimation with embedding vector features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 4233–4242.

Chen, W.; Jia, X.; Chang, H. J.; Duan, J.; Shen, L.; and Leonardis, A. 2021. FS-Net: Fast shape-based network for category-level 6D object pose estimation with decoupled rotation mechanism. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1581–1590.

Chen, Y.; Di, Y.; Zhai, G.; Manhardt, F.; Zhang, C.; Zhang, R.; Tombari, F.; Navab, N.; and Busam, B. 2024. Secondpose: Se (3)-consistent dual-stream feature fusion for category-level pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9959–9969.

Di, Y.; Zhang, R.; Lou, Z.; Manhardt, F.; Ji, X.; Navab, N.; and Tombari, F. 2022. GPV-Pose: Category-level Object Pose Estimation via Geometry-guided Point-wise Voting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 6781–6791.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask R-CNN. In *IEEE International Conference on Computer Vision*, 2961–2969.

He, Y.; Sun, W.; Huang, H.; Liu, J.; Fan, H.; and Sun, J. 2020. PVN3D: A deep point-wise 3d keypoints voting network for 6DoF pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 11632–11641.

Hinterstoisser, S.; Cagniart, C.; Ilic, S.; Sturm, P.; Navab, N.; Fua, P.; and Lepetit, V. 2011. Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5): 876–888.

Jung, H.; Wu, S.-C.; Ruhkamp, P.; Zhai, G.; Schieber, H.; Rizzoli, G.; Wang, P.; Zhao, H.; Garattoni, L.; Meier, S.; et al. 2024. HouseCat6D-A Large-Scale Multi-Modal Category Level 6D Object Perception Dataset with Household Objects in Realistic Scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22498–22508.

Kehl, W.; Manhardt, F.; Tombari, F.; Ilic, S.; and Navab, N. 2017. SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. In *IEEE International Conference on Computer Vision*, 1521–1529.

Lin, H.; Liu, Z.; Cheang, C.; Fu, Y.; Guo, G.; and Xue, X. 2022a. SAR-Net: Shape Alignment and Recovery Network for Category-Level 6D Object Pose and Size Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 6707–6717.

Lin, J.; Wei, Z.; Li, Z.; Xu, S.; Jia, K.; and Li, Y. 2021. Dualposenet: Category-level 6D object pose and size estimation using dual pose network with refined learning of pose consistency. In *IEEE International Conference on Computer Vision*, 3560–3569.

Lin, J.; Wei, Z.; Zhang, Y.; and Jia, K. 2023. VI-Net: Boosting Category-level 6D Object Pose Estimation via Learning Decoupled Rotations on the Spherical Representations. In *IEEE/CVF International Conference on Computer Vision*, 14001–14011.

Lin, S.; Wang, Z.; Ling, Y.; Tao, Y.; and Yang, C. 2022b. E2EK: End-to-End Regression Network Based on Keypoint for 6D Pose Estimation. *IEEE Robotics and Automation Letters*, 7(3): 6526–6533.

Lin, X.; Yang, W.; Gao, Y.; and Zhang, T. 2024. Instance-adaptive and geometric-aware keypoint learning for category-level 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21040–21049.

Liu, J.; Chen, Y.; Ye, X.; and Qi, X. 2023. IST-Net: Prior-Free Category-Level Pose Estimation with Implicit Space Transformation. In *IEEE/CVF International Conference on Computer Vision*, 13978–13988.

Manhardt, F.; Kehl, W.; and Gaidon, A. 2019. 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*, 2069–2078.

Oberweger, M.; Rad, M.; and Lepetit, V. 2018. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *European Conference on Computer Vision*, 119–134.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Peng, S.; Liu, Y.; Huang, Q.; Zhou, X.; and Bao, H. 2019. PVNet: Pixel-wise voting network for 6DoF pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 4561–4570.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30.

Rad, M.; and Lepetit, V. 2017. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. In *IEEE International Conference on Computer Vision*, 3828–3836.

Rad, M.; Oberweger, M.; and Lepetit, V. 2018. Feature mapping for learning fast and accurate 3d pose inference from synthetic images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 4663–4672.

Tian, M.; Ang, M. H.; and Lee, G. H. 2020. Shape prior deformation for categorical 6D object pose and size estimation. In *European Conference on Computer Vision*, 530–546.

Tjaden, H.; Schwanecke, U.; and Schomer, E. 2017. Real-time monocular pose estimation of 3D objects using temporally consistent local color histograms. In *Proceedings of the IEEE international conference on computer vision*, 124–132.

Wang, C.; Xu, D.; Zhu, Y.; Martín-Martín, R.; Lu, C.; Li, F.-F.; and Savarese, S. 2019a. Densefusion: 6D object pose estimation by iterative dense fusion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3343–3352.

Wang, H.; Sridhar, S.; Huang, J.; Valentin, J.; Song, S.; and Guibas, L. J. 2019b. Normalized object coordinate space for category-level 6D object pose and size estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2642–2651.

Wang, J.; Chen, K.; and Dou, Q. 2021. Category-level 6D object pose estimation via cascaded relation and recurrent reconstruction networks. In *IEEE International Conference on Intelligent Robots and Systems*, 4807–4814.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019c. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics*, 38(5): 1–12.

Wu, Y.; and Greenspan, M. 2024. Learning Better Keypoints for Multi-Object 6DoF Pose Estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 564–574.

Xiang, Y.; Schmidt, T.; Narayanan, V.; and Fox, D. 2018. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *Robotics: Science and Systems*.

Yu, S.; Zhai, D.-H.; and Xia, Y. 2024. CatFormer: Category-Level 6D Object Pose Estimation with Transformer. In *AAAI Conference on Artificial Intelligence*, volume 38, 6808–6816.

Zhang, R.; Di, Y.; Lou, Z.; Manhardt, F.; Tombari, F.; and Ji, X. 2022. RBP-Pose: Residual bounding box projection for category-level pose estimation. In *European Conference on Computer Vision*, 655–672.

Zheng, L.; Wang, C.; Sun, Y.; Dasgupta, E.; Chen, H.; Leonardis, A.; Zhang, W.; and Chang, H. J. 2023. HS-Pose: Hybrid Scope Feature Extraction for Category-level Object Pose Estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17163–17173.