2025 IEEE International Conference on Robotics and Automation (ICRA)
May 19-23, 2025. Atlanta, USA

A Unified End-to-end Network for Category-level and Instance-level Object Pose Estimation from RGB Images

Jiale Ren, Hong Liu*, Jinfu Liu, Peifeng Jiang

Abstract— **Accurately estimating the 6-DoF pose of objects is a fundamental challenge in computer vision and robotics. While category-level pose estimation based on RGBD data has achieved good performance in recent years, estimating poses solely from RGB images remains a significant challenge. Existing RGB-based category-level methods primarily focus on recovering object point clouds from RGB images, and pose prediction is not performed end-to-end by a network. This paper presents a Category-level and Instance-level Pose Estimation Network (CIPE), which models pose estimation as a set prediction problem and enables direct pose regression from RGB images. To further enhance the network's ability to learn object poses, first, a novel learnable rotation representation that redefines rotation learning within Euclidean space is introduced to facilitate rotation regression. Additionally, we propose a prior-query fusion strategy that utilizes a pre-trained point cloud feature extraction network to integrate categorical object features with bounding boxes, thereby improving the incorporation of category information. Experimental results demonstrate that CIPE significantly outperforms existing RGB-based methods on both category-level and instance-level datasets. The code is available at https://github.com/jialeren/CIPE.**

I. INTRODUCTION

6-DoF (degrees-of-freedom) object pose estimation aims to find the rigid body transformation from object frame to camera frame. As a fundamental problem in the field of computer vision and robotics, it is crucial for various applications, including manipulation, industrial assembly, autonomous driving, and augmented reality.

The task of pose estimation can be divided into two types: instance-level and category-level pose estimation. The former estimates the pose of specific instances using their CAD model, whereas the latter can predict the pose for objects within a certain category. Instance-level methods typically calculate pose by establishing correspondences between 2D pixels in the image and 3D points of the CAD model, followed by the PnP algorithm [1]. Although achieving high accuracy, their applicability is limited in real-world scenarios due to the difficulty of obtaining CAD models. Conversely, category-level methods eliminate the need for CAD models and solve the problem based on similarity transformation between the point cloud from the camera's perspective and the point cloud defined in the normalized object coordinate space (NOCS) [2], which represents different instances within a category in one unified space. Different instances of the same class may be of different sizes, causing scale ambiguity. Therefore, category-level methods often match point cloud coordinates by Umeyama algorithm [3] to solve the pose and size of objects. The two different paradigms of instance-level and category-level pose estimation determine that existing methods employ distinct approaches for these two tasks.

Furthermore, reconstructing point cloud requires depth information, as most category-level methods rely on RGBD data. Despite advancements in depth sensors, it is more costly and less accessible compared to RGB images in many scenarios like wearable AR and indoor robotics. While instance-level methods can operate with only RGB images, there is a notable lack of focus on RGB-based category-level object pose estimation. Current RGB-based methods [4]–[7] mainly also adapt RGBD-based approaches when depth information is unavailable, they overlook the potential of directly estimating pose from RGB pixels, limiting the networks' ability to extract spatial information from images.

To address the aforementioned issues, we propose a more direct approach to solve the pose estimation problem. Our motivation stems from the fact that object detection models can identify objects of a particular category in RGB images without restricting the instances of objects [8], [9]. Therefore, we aim to develop an end-to-end model capable of estimating poses directly from RGB images, whether at the instance or category level. To achieve this, we propose a novel approach named CIPE, which stands for category-level and instance-level pose estimation. Drawing on insights from instance-level end-to-end method PoET [10], CIPE extends the paradigm of set prediction and employs a DETR-based [5] network for end-to-end pose regression. Through end-to-end learning, the network autonomously learns pose information without the need for manually prescribed algorithms, unifying instance-level and category-level pose estimation into a single problem. To the best of our knowledge, CIPE is the first network to employ an end-to-end paradigm for both category-level and instance-level pose estimation.

One of the key challenges in implementing this end-to-end approach is the inability to directly use the rotation matrix defined on $SO(3)$ as the network's learning target, which hinders the network's capacity for learning rotations effectively. This work introduces a new learnable rotation representation that enables the model to calculate the rotation loss directly in Euclidean space. Furthermore, the integration of a pre-trained point cloud feature extraction model into

*Corresponding Author (e-mail: hongliu@pku.edu.cn)
All authors are affiliated with the State Key Laboratory of General Artificial Intelligence, Peking University, Shenzhen Graduate School, China, and the Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology, Peking University, Shenzhen Graduate School, China.

This work was supported in part by the National Natural Science Foundation of China under Grant 62373009 and in part by the Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology under Grant 2024B1212010006.

2025 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3315-4139-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICRA55743.2025.11128247

our framework facilitates the utilization of categorical object priors to aid in pose estimation. Building on these improvements, CIPE takes only RGB images as input, enabling both category-level and instance-level pose estimation to be performed in an end-to-end manner. In summary, this paper makes the following contributions:

- We provide a unified method that estimates category-level and instance-level object poses solely from single RGB images. Through end-to-end learning, our approach achieves significant improvements over existing RGB-based methods.
- A novel learnable rotation representation is introduced, which operates directly in Euclidean space to enhance the continuity and accuracy of rotation estimation, offering a more effective approach compared to traditional representations.
- A prior-query fusion module is introduced to extract point cloud information as a category prior, aiding the model in understanding categorical information.

II. Related Work

A. Instance-level Object Pose Estimation

Mainstream instance-level pose estimation methods simplify the pose estimation process by first predicting key points and then solving the pose using the PnP algorithm [11]–[13]. Alternatively, there's a growing trend towards end-to-end networks for pose estimation. PoseCNN [14] decouples translation and rotation and uses a CNN to predict them separately. REDE [15] and YoLoPose v2 [16] make an effort to make the PnP process differentiable. In recent years, the introduction of various vision backbone networks based on Transformers [17] has further improved the performance of end-to-end pose estimation. T6D-Direct [18] and PoET [10] incorporate the DETR [9] network into object pose estimation, modeling the problem as a set prediction task and achieving an end-to-end approach using only RGB images as input. This framework blurs the distinction between instance-level and category-level pose estimation, showing potential for unified pose estimation.

B. Category-level Object Pose Estimation

1) RGBD-based: The majority of category-level pose estimation methods rely on RGBD data. Wang *et al.* [2] pioneer this task by introducing normalized object coordinate space (NOCS), using the Umeyama [3] algorithm to determine the size and pose of the object by predicting the NOCS map and matching it with the depth map. Many subsequent methods [19]–[21] follow this paradigm. More recent approaches include the use of diffusion models [22] and the integration of large language models [23]. Despite excellent performance through various techniques, all these methods require the restoration of point clouds from RGBD images and do not directly predict the object pose.

2) RGB-based: Recovering point clouds without depth information is challenging, making it difficult for RGB-based methods to directly use RGBD frameworks. Consequently, few studies have focused on RGB-based category-level pose estimation. Chen *et al.* [24]propose an analysis-by-synthesis method that utilizes a parametric image synthesis module to recover object pose. Lee *et al.* [4] introduce a dual-branch framework aimed at estimating metric scale shape and pose. OLD-Net [5] predicts object-level depth from RGB images by deforming category-level shape priors. Lastly, [6] employs a decoupled framework that separates the estimation of 6D pose and size, while FAP-Net [7] utilizes a coarse-to-fine framework for end-to-end category-level pose estimation. However, these methods still follow the paradigm of predicting depth and reconstructing object point clouds. In contrast, our approach leverages visual networks to regress object poses directly from images within an end-to-end framework, bypassing the need for shape, appearance, depth, or point clouds, and eliminating handcrafted pose-solving algorithms.

III. METHODS

A. Overview

The architecture of CIPE is depicted in Fig. 1. Inspired by DETR [9] and PoET [10], CIPE treats pose estimation as a set prediction problem. First, an RGB image is fed into a Mask R-CNN [8] backbone to obtain multi-scale feature maps and object detection bounding boxes. These feature maps are then processed through a convolution layer to adjust the dimension, and position encoding is employed to the four vertex coordinates of the object detection box. The center coordinates of the bounding box after position encoding are used as the reference points, and the vertex coordinates are fed into our proposed prior-query fusion (PQF) module to get object queries with prior information, which are forwarded to an original Deformable DETR [25] decoder together with the projected multi-scale image features. Finally, the decoder produces as many outputs as target classes, and each of these outputs is directed to a prediction head to obtain the final predicted pose. Each part will be introduced in detail below.

B. Network Design

Given an input RGB image, a pre-trained Mask R-CNN network is first used for object detection and feature extraction. The output consists of a series of bounding boxes $\{(x_1, y_1, w_1, h_1), \cdots, (x_i, y_i, w_i, h_i)\}$ and multi-scale features $\{F_1, F_2, F_3\}$. After a convolutional layer, these multi-scale features are then flattened and concatenated to multi-scale feature F, which is used as input to the subsequent decoder. According to the prediction box and our proposed prior query fusion (PQF) module, a set of object queries Q and reference points C can be obtained. Subsequently, a decoder consists of Transformer decoding layers with deformable attention mechanism takes F,Q,C as input and outputs a set of results X with the same dimensions as Q. For each element X_i in X, a shared-weight prediction head directly predicts rotation R, translation t, and object category c in regression manner. For category-level pose estimation, CIPE doesn't use a separate module; instead, it simply adds a size prediction branch in the prediction head. Therefore, our model supports both instance-level and category-level

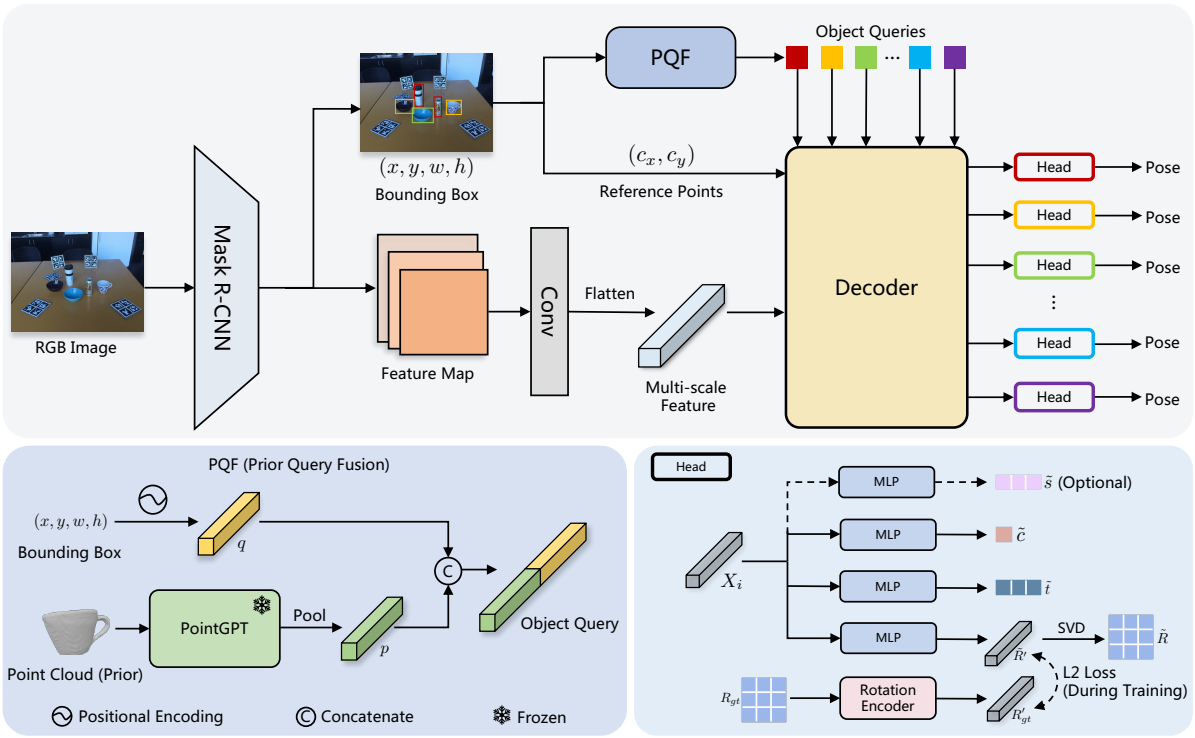

Fig. 1. An overview of CIPE. Given an RGB image as input, CIPE initially performs object detection and feature extraction by Mask R-CNN. The PQF module combines the detected object bounding boxes with the category-level features extracted from a pre-trained model to generate object queries containing prior information. These queries are then processed by a decoder, and a set of shared-weight prediction heads outputs the estimated rotation r, translation t, class c, and size s, respectively. For the prediction of rotation, our proposed learnable rotation representation is used during training.

pose estimation with no structural differences, except for the added size prediction branch.

Additionally, DETR's architecture is primarily tailored for object detection, but in 6D pose estimation, Mask R-CNN already provides bounding boxes and multi-scale features. Replicating the encoder-decoder structure would be unnecessary. Through experiments, we found that pose regression is more effective when directly leveraging Mask R-CNN's features, thus CIPE excludes the Transformer encoder.

C. Learnable Rotation Representation

Previous studies [26], [27] have highlighted the shortcomings of traditional rotation representations such as Euler angles and quaternions in rotation regression tasks. Experimental results demonstrate that Procrustes mapping [28] offers superior performance in rotation learning. Considering a Procrustes problem: given a matrix $M \in \mathbb{R}^{3\times3}$, the objective is to find the nearest rotation matrix R:

$$\underset{R\in SO(3)}{\arg\min} ||R - M||_F^2, \tag{1}$$

and its solution is:

$$\text{Procrustes}(M) = UV^T, \tag{2}$$

where $SVD(M) = USV^T$ and UV^T must be a rotation matrix. Therefore, a straightforward approach is to add an SVD decomposition operation after the network's final layer output to obtain a rotation matrix and compute the loss in $SO(3)$. However, the neural network's output is inherently defined in Euclidean space, hence we aim to compute the loss directly in Euclidean space.

To achieve this, we consider constraining M directly in the Euclidean space before SVD decomposition. As illustrated in Fig. 2, we design an autoencoder network where the rotation matrix R is mapped to a 3×3 matrix in the Euclidean space through a Multi-Layer Perceptron (MLP). An SVD decomposition is then performed on this matrix, yielding $Y = UV^T$, which serves as the network's output. In this network, the intermediate variable R' represents a rotation matrix in Euclidean space. Through the trained network, the position encoding and MLP parts are unified as $Encoder_{rot}$ (Rotation Encoder), satisfying the relationship:

$$\text{Procrustes}(Encoder_{rot}(R)) \approx R, \tag{3}$$

where R is an arbitrary rotation matrix. Unlike mathematical representations, $Encoder_{rot}$ may not mathematically represent a strict rotation matrix, it provides a continuous and unique representation that improves the learning capabilities of rotation regression. In CIPE, rotation labels in Euclidean space are derived using this rotation auto-encoder network. Specifically, for the ground-truth rotation matrix label R_{gt} in the dataset, $R'_{gt} = Encoder_{rot}(R)$ is used as the label in Euclidean space. Since $Encoder_{rot}$ is learned through a network, we refer to this approach as a learnable rotation representation. Subsequent experiments will demonstrate that the errors inherent to the autoencoder training do not degrade pose estimation results; on the contrary, direct regression in Euclidean space achieves superior results compared to other

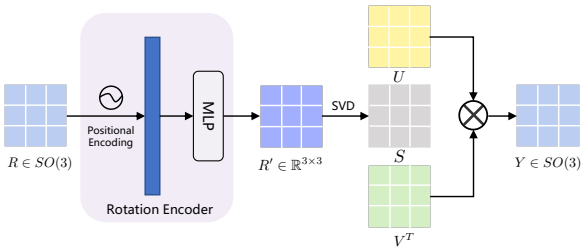

Fig. 2. An auto-encoder network transforms the rotation matrix by projecting it into a 128-dimensional vector after position encoding. Subsequently, it traverses through an MLP comprising two linear layers and a LeakyReLU activation function. Finally, SVD decomposition is used to obtain the rotation matrix.

rotation representations, including the 6D representation [26] and Procrustes mapping [27], [28].

D. Prior Query Fusion

Inspired by PoET [10], we utilize the coordinates of the predicted bounding boxes instead of learnable tokens as object queries for the decoder. For a given prediction box (x, y, w, h), sinusoidal position encoding is applied to transform it into a higher-dimensional vector of length L:

$$q = \text{PE}((x, y, w, h)) \in \mathbb{R}^L. \tag{4}$$

Besides, a pre-trained model PointGPT [29] is employed for point cloud feature extraction. Given point cloud P, PointGPT embeds it into D-dimensional tokens:

$$T = \text{PointGPT}(P) \in \mathbb{R}^{n\times D}. \tag{5}$$

For instance-level objects, P is obtained from the CAD model with average pooling on T. For category-level objects, T is averaged across N objects in each category. A linear layer projects these prior features to match q's dimension, yielding feature p:

$$p = \begin{cases} \text{FC}(\text{Pool}(T)) \in \mathbb{R}^L, & \text{instance-level,} \\ \text{FC}(\text{Pool}(\frac{1}{N}\sum_{i=1}^{N} T_i)) \in \mathbb{R}^L, & \text{category-level.} \end{cases} \tag{6}$$

The box query q and category prior feature p are concatenated as an object query. For multiple bounding boxes detected by Mask R-CNN in each image, individual queries Q_i are computed:

$$Q_i = \text{Concat}(p, q). \tag{7}$$

The decoder has a preset number of object queries N_q. If the number of detected bounding boxes N_b is less than N_q, we pad the remaining queries with zeros. Thus, for each input image, the composition of object queries is:

$$Q = [Q_1, \cdots, Q_{N_b}, \mathbf{0} \times (N_q - N_b)]. \tag{8}$$

Note that PQF is used to further assist in understanding object category information when object models are available. For category-level pose estimation, the average of all instances within the same object category is used as the object prior, and the exact CAD model of the object is not required during inference. In more extreme cases, CIPE can still work effectively in an RGB-only manner, even without the PQF module.

E. Loss Functions

In the output head of CIPE, the translation head outputs the translation $\tilde{t}$, and the rotation head outputs the rotation prediction $\tilde{R}'$ in our learnable rotation representation space. Given the ground-truth translation t_{gt} and rotation matrix R_{gt}, L2 loss is employed as translation and rotation loss:

$$L_{trans} = ||t_{gt} - \tilde{t}||_2, \tag{9}$$

$$L_{rot} = ||Encoder_{rot}(R_{gt}) - \tilde{R}'||_2. \tag{10}$$

Cross-entropy loss and L1 loss are used to calculate the classification loss L_{label} and size loss L_{size}, respectively. When training the network, the four components of the loss function are weighted together:

$$L = \lambda_1 L_{trans} + \lambda_2 L_{rot} + \lambda_3 L_{label} + \lambda_4 L_{size}, \tag{11}$$

where λ_1 to λ_4 are the weighting parameters.

IV. Experiments and Discussion

A. Datasets

1) CAMERA25 and REAL275: For category-level pose estimation, the two most commonly used datasets are CAMERA25 and REAL275 [2]. CAMERA25 contains 300,000 images synthesized by context-aware mixed reality method. The synthesized object poses comply with objective physical laws and are close to real scenes. REAL275 is completely collected in real scenes, with a total of 8,000 images.

2) LM-O: The LM-O [30] (LINEMOD-Occluded) benchmark is a standard and challenging dataset for instance-level pose estimation, designed with scenes where objects occlude each other. It encompasses 8 categories and contains a total of 47,702 RGB images, with both real and rendered objects.

B. Implementation Details

On the CAMERA25 dataset, training CIPE involves 5 epochs using the AdamW [31] optimizer, with a learning rate of 2×10^{-5} and batch size of 32, all executed on a single NVIDIA 3090 GPU. The CIPE architecture consists of 6 decoder layers with N_q = 9, d_h = 256, 16 attention heads, and a positional embedding dimension of $L = 32$. This network is trained end-to-end with weighting parameters set $\lambda_1 = \lambda_2 = \lambda_3 = 1, \lambda_4 = 10$ to maintain losses within the same order of magnitude. When extracting point cloud features in prior-query fusion, we use the feature extractor part of the PointGPT-L with the weights pre-trained on ShapeNet [32]. On the REAL275 dataset, using the same settings, fine-tuning the weights trained on the CAMERA25 for 5 epochs yields satisfactory results. The LM-O dataset is relatively small, necessitating 50 epochs of training while retaining the same parameter settings.

C. Evaluation Metrics

The most common metric of category-level pose estimation is $n°, n$ cm, On CAMERA25 and REAL275 dataset, we report the $10°, 10$ cm for pose estimation. In addition, the mean average precision (mAP) of 3D IoU under different thresholds is also reported for pose and size estimation.

TABLE I

COMPARISON WITH OTHER CATEGORY-LEVEL METHODS ON CAMERA25 AND REAL275 DATASET.

Method	CAMERA25					REAL275				
	$3D_{50}$	$3D_{75}$	10°	10cm	10°,10cm	$3D_{50}$	$3D_{75}$	10°	10cm	10°,10cm
Synthesis (ECCV'20) [24]	-	-	-	-	-	-	-	14.2	34.0	4.8
MSOS (RAL'21) [4]	32.4	5.1	60.8	29.7	19.2	23.4	3.0	29.2	39.5	9.6
OLD-Net (ECCV'22) [5]	32.1	5.4	74.0	30.1	23.4	25.4	1.9	37.0	38.9	9.8
DMSR (ICRA'24) [6]	34.6	6.5	81.4	32.3	27.4	28.3	6.1	59.5	37.3	23.6
FAP-Net (ICRA'24) [7]	39.2	6.7	80.4	36.0	29.8	36.8	5.2	49.6	49.7	24.5
CIPE (Ours)	**54.9**	**19.5**	**93.9**	**53.5**	**51.8**	**71.7**	**22.1**	**68.6**	**85.0**	**60.1**

For the LM-O dataset, ADD-(S) is employed as evaluation metric that considers the error between point clouds of objects, with a specific focus on symmetric objects. The calculation formula is as follows:

$$\text{ADD-(S)} = \begin{cases} \frac{1}{|M|}\sum\limits_{x_1 \in M}\min\limits_{x_2 \in M}\|(\tilde{R}x_1+\tilde{t})-(Rx_2+t)\|, & \text{if symmetry,} \\ \frac{1}{|M|}\sum\limits_{x \in M}\|(\tilde{R}x+\tilde{t})-(Rx+t)\|, & \text{otherwise.} \end{cases} \tag{12}$$

where M denotes the set of 3D model points.

D. Comparison with State-of-the-art

1) Category-level Results: The performance comparison between CIPE and existing RGB-based methods is shown in Table I. It can be seen that CIPE significantly outperforms the state-of-the-art RGB-based method DMSR [6], showing improvements of 24.4% on CAMERA25 and 36.5% on REAL275 in the key metric 10°, 10 cm. In some metrics, CIPE even more than doubles the performance. This is attributed to its end-to-end pose prediction approach, which eliminates inaccuracies that arise from recovering depth from RGB images. Learning poses directly from RGB pixels offers another advantage: unlike other methods that require reconstructing point clouds from RGB images, which can introduce gaps between datasets, the end-to-end learning process avoids this issue. As a result, CIPE achieves strong performance on the REAL275 dataset as well.

Fig. 3 provides a detailed analysis of average precision (AP) at different thresholds, which demonstrates the performance across different object categories in terms of rotation error, translation error, and 3D IoU. We also present the results of DMSR [6], demonstrating the performance advantages of CIPE in various metrics. Notably, the AP curves for each category within the CAMERA25 dataset exhibit greater concentration compared to those within the REAL275 dataset. This discrepancy may stem from CAMERA25's utilization of synthesized data, resulting in less pronounced scene variations within the dataset. Furthermore, the model performs poorly in predicting the rotation of the camera across two datasets due to the greater intra-class variability within the camera category compared to others.

2) Instance-level Results: The performance comparison of CIPE with other methods on the LM-O dataset is presented in Table II. From the results, it is evident that CIPE demonstrates superior performance compared to the existing state-of-the-art end-to-end method PoET [10]. Additionally, CIPE also achieves better or on-par performance when compared with the two-stage method. Note that these two-stage methods require the use of CAD models, while CIPE only takes RGB images as input at the inference stage, which validates the introduction of our learnable rotation representation and the prior-query fusion method.

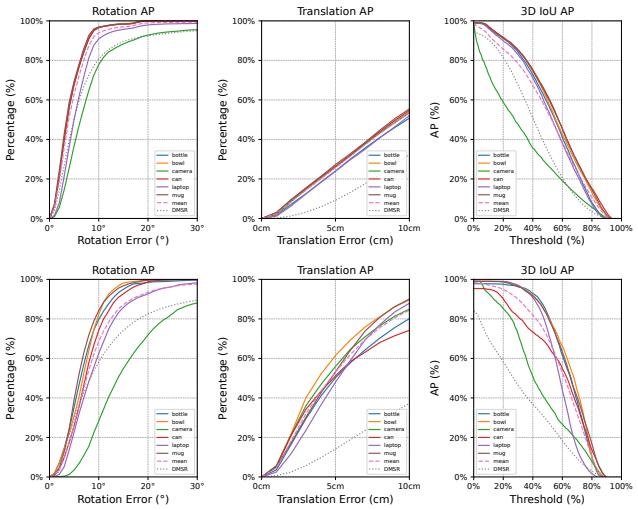

Fig. 3. Upper: Results on CAMERA25 dataset; Lower: Results on REAL275 dataset, both showing average precision (AP) with different thresholds for rotation error, translation error, and 3D IoU. DMSR refers to the method proposed in [6].

E. Ablation Study

1) Different Design Choice: To investigate the effectiveness of our proposed learnable rotation representation (LR), prior query fusion (PQF), as well as the influence of remove encoder (RM). We use PoET (with 6D representation) as baseline and design four experimental conditions: LR only, RM only, PQF only, and the combined LR-RM-PQF (CIPE). As shown in the upper part of Table III, each factor enhances pose estimation performance to varying degrees across different datasets. When all three factors are applied together, the model shows substantial improvements in all evaluation metrics compared to the baseline.

2) Effectiveness of Learnable Rotation Representation: We first conduct experiments akin to [26] by performing self-regression of rotation matrices. An MLP is employed to learn the rotation matrices, utilizing various mathematical representations as the output of the final layer. As shown in Table IV, despite the error in the rotation encoder, our proposed learnable representation achieves comparable results with both 6D and Procrustes representation. Although the learnable rotation representation does not yield the best results, it remains an effective approach, which is exactly the issue we aim to highlight: rather than being a strict

TABLE II

COMPARISON WITH OTHER INSTANCE-LEVEL METHODS WITH ADD-(S) ON LM-O DATASET.

Method	End-to-end (RGB-based)			Two-stage (with CAD Models)					
	PoseCNN [14]	PoET [10]	CIPE (Ours)	Pix2Pose [33]	HybridPose [34]	PVNet [11]	GDR-Net [35]	CDPN [12]	RNNPose [36]
Ape	9.6	12.7	**44.4**	22.0	20.9	15.8	39.3	59.2	37.2
Can	45.2	51.0	**65.4**	44.7	75.3	63.3	79.2	63.5	88.1
Cat	0.93	10.9	**20.3**	22.7	24.9	16.7	23.5	26.2	29.2
Driller	41.4	53.3	**80.0**	44.7	70.2	65.7	71.3	55.6	88.1
Duck	19.6	22.6	**50.9**	15.0	27.9	46.9	44.4	52.4	49.2
Eggbox	22.0	50.4	**58.2**	25.2	52.4	54.2	58.2	63.0	67.0
Glue	38.5	63.9	**81.3**	32.4	53.8	75.8	49.3	71.7	63.8
Holep.	22.1	29.8	**57.0**	49.5	54.2	36.1	58.7	52.5	62.8
Mean	24.9	36.8	**57.2**	32.0	47.5	40.8	53.0	55.5	60.7

TABLE III

THE UPPER PART IS THE IMPACT OF THE PRESENCE OF REPRESENTATION (LR), REMOVE ENCODER (RM), AND PRIOR QUERY FUSION (PQF) ON THE MODEL PERFORMANCE; THE LOWER PART IS THE IMPACT OF DIFFERENT ROTATION PREFERENCES ON THE MODEL PERFORMANCE.

RM	LR	PQF	LM-O			CAMERA25			REAL275		
			ADD-(S)	T. Error (cm)	R. Error (°)	10°	10cm	10°,10cm	10°	10cm	10°,10cm
			29.5	3.3	47.6	80.1	49.3	43.5	63.4	78.9	52.0
✓			39.4	3.0	29.9	87.9	50.3	44.9	64.6	82.0	54.4
	✓		44.2	3.0	32.7	92.5	52.1	50.1	66.1	79.2	53.6
		✓	41.9	3.0	30.7	93.0	49.4	47.1	68.1	84.4	54.5
Quaternion			6.4	6.7	71.9	63.8	51.9	35.2	23.7	55.2	13.2
6D			34.9	3.2	38.8	85.1	51.8	45.4	64.7	82.1	56.0
Procrustes			34.7	3.1	36.2	85.5	53.9	47.1	65.7	83.5	56.7
CIPE (Ours)			**57.2**	**2.3**	**28.1**	**93.9**	**53.5**	**51.8**	**68.6**	**85.0**	**60.1**

TABLE IV

DEGREE ERRORS(°) IN SELF-REGRESSION EXPERIMENT ACROSS 4 ROTATION REPRESENTATIONS.

Representation	Min(°)	Max(°)	Mean(°)
Quaternions	0.08	71.24	1.52
6D	0.05	1.78	0.49
Procrustes	0.05	1.11	0.38
Learnable (Ours)	0.05	1.70	0.68

rotation representation, it serves as a suitable target for neural networks to learn rotation.

To validate its effectiveness in pose estimation, we further use different rotation representations as learning targets. The results are shown in the lower part of Table III. It is evident that, for both category-level and instance-level pose estimation, the learnable rotation representation yields significantly superior results compared to other representations. Although it is not a lossless representation, allowing the network to perform regression in Euclidean space effectively improves the performance of end-to-end pose estimation.

F. Qualitative Results and Limitations

Fig. 4 presents a visualization of some pose estimation results, demonstrating that CIPE accurately predicts the poses of objects with varying styles within the same category in complex scenes, across both the CAMERA25 and REAL275 datasets. In addition, in certain cases where translation predictions are less accurate (e.g., the bottle in the first image of the second row and the laptop in the fourth image of the second row, marked by the yellow boxes), the predicted coordinate axis center is slightly offset from the real coordinate. This indicates that our method, which does not include a dedicated design for translation prediction, could benefit from further refinement in translation accuracy.

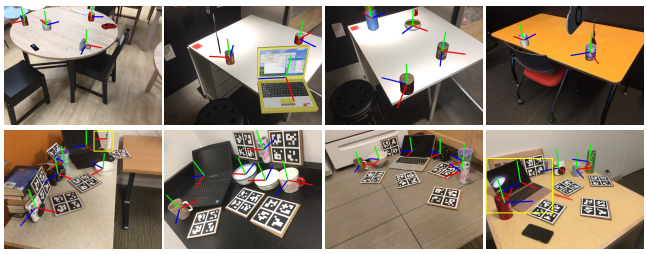

Fig. 4. The visualization presents the predicted poses. The upper half displays the results from the CAMERA25 dataset, while the lower half shows those from the REAL275 dataset. The blue, green, and red lines represent the X, Y, and Z axis, respectively. Yellow boxes mark cases where translation predictions are inaccurate.

V. CONCLUSIONS

This paper introduces an RGB-based object pose estimation network capable of predicting both instance-level and category-level poses in an end-to-end manner, complemented by a novel learnable rotation representation and a prior query fusion module. By leveraging end-to-end training from RGB pixels, our approach fully utilizes the network's spatial representation capabilities without relying on depth information or manually designed coordinate matching and refinement processes. Extensive experimental results validate the effectiveness of our method in regressing rotations and translations. The promising results demonstrate that our approach bridges the gap between RGB and RGBD-based methods, highlighting the feasibility of using RGB images and end-to-end networks for pose estimation.

REFERENCES

[1] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[2] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2642–2651.

[3] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 04, pp. 376–380, 1991.

[4] T. Lee, B.-U. Lee, M. Kim, and I. S. Kweon, "Category-level metric scale object shape and pose estimation," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8575–8582, 2021.

[5] Z. Fan, Z. Song, J. Xu, Z. Wang, K. Wu, H. Liu, and J. He, "Object level depth reconstruction for category level 6d object pose estimation from monocular rgb image," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 220–236.

[6] J. Wei, X. Song, W. Liu, L. Kneip, H. Li, and P. Ji, "Rgb-based category-level object pose estimation via decoupled metric scale recovery," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2036–2042.

[7] J. Li, L. Jin, X. Song, Y. Chen, N. Li, and X. Qin, "Implicit coarse-to-fine 3d perception for category-level object pose estimation from monocular rgb image," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2043–2050.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.

[9] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 213–229.

[10] T. G. Jantos, M. A. Hamdad, W. Granig, S. Weiss, and J. Steinbrener, "Poet: Pose estimation transformer for single-view, multi-object 6d pose estimation," in *Conference on Robot Learning (CoRL)*, 2023, pp. 1060–1070.

[11] S. Peng, X. Zhou, Y. Liu, H. Lin, Q. Huang, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof object pose estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3212–3223, 2022.

[12] Z. Li, G. Wang, and X. Ji, "Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 7678–7687.

[13] G. Billings and M. Johnson-Roberson, "Silhonet: An rgb method for 6d object pose estimation," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3727–3734, 2019.

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

[15] W. Hua, Z. Zhou, J. Wu, H. Huang, Y. Wang, and R. Xiong, "Rede: End-to-end object 6d pose robust estimation using differentiable outliers elimination," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2886–2893, 2021.

[16] A. S. Periyasamy, A. Amini, V. Tsaturyan, and S. Behnke, "Yolopose v2: Understanding and improving transformer-based 6d pose estimation," *Robotics and Autonomous Systems (RAS)*, vol. 168, p. 104490, 2023.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[18] A. Amini, A. S. Periyasamy, and S. Behnke, "T6d-direct: Transformers for multi-object 6d pose direct regression," in *DAGM German Conference on Pattern Recognition*, 2021, pp. 530–544.

[19] M. Tian, M. H. Ang, and G. H. Lee, "Shape prior deformation for categorical 6d object pose and size estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 530–546.

[20] K. Chen and Q. Dou, "Sgpa: Structure-guided prior adaptation for category-level 6d object pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2773–2782.

[21] L. Zou, Z. Huang, N. Gu, and G. Wang, "6d-vit: Category-level 6d object pose estimation via transformer-based instance representation learning," *IEEE Transactions on Image Processing*, vol. 31, pp. 6907–6921, 2022.

[22] J. Zhang, M. Wu, and H. Dong, "Generative category-level object pose estimation via diffusion models," *Advances in Neural Information Processing Systems(NeurIPS)*, vol. 36, 2024.

[23] X. Lin, M. Zhu, R. Dang, G. Zhou, S. Shu, F. Lin, C. Liu, and Q. Chen, "Clipose: Category-level object pose estimation with pre-trained vision-language knowledge," *arXiv preprint arXiv:2402.15726*, 2024.

[24] X. Chen, Z. Dong, J. Song, A. Geiger, and O. Hilliges, "Category level object pose estimation via neural analysis-by-synthesis," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 139–156.

[25] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations (ICLR)*, 2021.

[26] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5745–5753.

[27] R. Brégier, "Deep regression on manifolds: a 3d rotation case study," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 166–174.

[28] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[29] G. Chen, M. Wang, Y. Yang, K. Yu, L. Yuan, and Y. Yue, "Pointgpt: Auto-regressively generative pre-training from point clouds," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2024.

[30] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6d object pose estimation using 3d object coordinates," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014, pp. 536–551.

[31] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, 2018.

[32] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

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

[34] C. Song, J. Song, and Q. Huang, "Hybridpose: 6d object pose estimation under hybrid representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 431–440.

[35] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16 611–16 621.

[36] Y. Xu, K.-Y. Lin, G. Zhang, X. Wang, and H. Li, "Rnnpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 14 880–14 890.