

You Only Pose Once: A Minimalist’s Detection Transformer for Monocular RGB Category-level 9D Multi-Object Pose Estimation

Hakjin Lee, Junghoon Seo, Jaehoon Sim

Abstract—Accurately recovering the full 9-DoF pose of unseen instances within specific categories from a single RGB image remains a core challenge for robotics and automation. Most existing solutions still rely on pseudo-depth, CAD models, or multi-stage cascades that separate 2D detection from pose estimation. Motivated by the need for a simpler, RGB-only alternative that learns directly at the category level, we revisit a longstanding question: *Can object detection and 9-DoF pose estimation be unified with high performance, without any additional data?* We show that they can be achieved with our method, YOPO, a single-stage, query-based framework that treats category-level 9-DoF estimation as a natural extension of 2D detection. YOPO augments a transformer detector with a lightweight pose head, a bounding-box-conditioned translation module, and a 6D-aware Hungarian matching cost. The model is trained end-to-end only with RGB images and category-level pose labels. Despite its minimalist design, YOPO sets a new state of the art on three benchmarks. On the REAL275 dataset, it achieves 79.6% IoU₅₀ and 54.1% under the 10°10cm metric, surpassing all prior RGB-only methods and closing much of the gap to RGB-D systems. The code, models, and additional qualitative results can be found on our project page¹.

I. INTRODUCTION

The ability to determine an object’s three-dimensional position and orientation, known as 6D pose estimation, is a cornerstone of robotic intelligence, enabling critical applications in robotic manipulation [1], [2], [3], augmented reality [4], [5], and autonomous driving [6], [7]. While early research focused on estimating the pose of specific, known object instances [8], [9], [10], the practical need to handle novel objects has driven a shift towards category-level pose estimation [11], [12], [13]. This more challenging task aims to generalize to previously unseen objects within a given category rather than to only those seen during training.

This work focuses on the particularly challenging and practical setting of **monocular RGB, category-level, multi-object pose estimation**. Relying on a single RGB image makes this approach highly accessible and cost-effective compared with methods that require active depth sensors [14], [15]. However, the lack of explicit 3D information introduces significant ambiguity in both object depth and scale. Consequently, the task expands to estimating a 9-Degree-of-Freedom (9-DoF) pose, comprising not only the 3D rotation $R \in SO(3)$ and 3D translation $t \in \mathbb{R}^3$ but also the object’s metric 3D size $s \in \mathbb{R}_{>0}^3$ to account for intra-class shape variations [16], [17]. The ultimate goal of this task is to develop a model and pipeline that, given an

	CAD Model?	Instance Mask?	Pseudo-depth?
Others	[15], [18], [16]	[19], [17], [20]	[14], [1], [18]
Ours	X	X	X

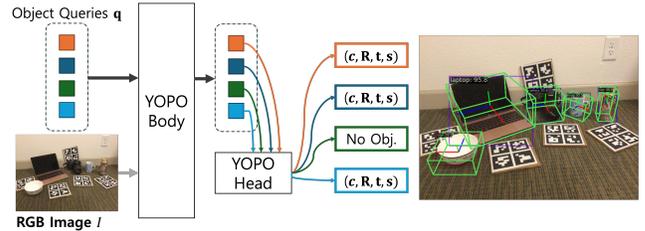


Fig. 1: **Main contribution of this paper.** Unlike prevailing category-level pose estimation methods that rely on external geometric priors such as 3D CAD models, instance segmentation masks, or pseudo-depth maps (top), our framework is end-to-end and requires none of these (bottom). Using only a raw RGB image as input, YOPO delivers state-of-the-art joint detection and 9D pose estimation for all objects in a single forward pass, with no intermediate steps or post-processing.

RGB image $I_i \in \mathbb{R}^{H \times W \times 3}$, can detect and estimate the pose of all objects present, outputting a set of predictions $\{c_j, R_j, t_j, s_j\}_{j=1}^{M_i}$, where c_j is the object class and M_i is the number of observed objects.

Despite significant progress, most leading approaches are not truly end-to-end. Instead, they rely on complex, multi-stage pipelines. Furthermore, these methods often require auxiliary data. The data usually include category-specific shape priors from 3D CAD models [15], [21], [19], instance segmentation masks for initial object localization [14], [1], [16], or estimated pseudo-depth maps to simplify 3D reasoning [14], [18]. Such dependencies hinder end-to-end training, increase computational overhead, and create performance bottlenecks that depend on these external modules.

This paper questions the necessity of such complex pipelines. We draw inspiration from the success of the modern query-based detection transformer (DETR) [22], [23], which has demonstrated the power of formulating detection as a direct set prediction problem. We investigate whether

AI Robot Team, PIT IN Co., Anyang-si, Gyeonggi-do, Republic of Korea.
Email: {hj, sjh, simjeh}@pitin-ev.com

¹<https://mikigom.github.io/YOPO-project-page>

this end-to-end paradigm can be extended from 2D detection to the challenging 9-DoF 3D pose estimation domain. To this end, we introduce **YOPO**: a new framework for monocular, category-level 9D pose estimation that operates in a truly end-to-end fashion. YOPO is built upon a transformer-based object detector and learns to directly predict an object’s bounding box, class, 3D rotation, 3D translation, and 3D scale in a single forward pass. It is trained jointly using only raw RGB images and their corresponding category-level 9D pose annotations. Crucially, YOPO dispenses with the need for 3D CAD models, shape priors, instance segmentation masks, pseudo-depth maps, or even explicit 2D bounding boxes during the training and inference stages. Our experiments show that this simpler, more direct approach not only streamlines the process but also establishes a new state of the art on standard benchmarks.

As shown in Fig. 1, our main contributions are as follows:

- We propose YOPO, a novel single-stage, query-based framework for monocular category-level 9D object pose estimation that is fully end-to-end trainable and requires only RGB images and 9D pose labels.
- We present a minimalist yet effective design that augments a detection transformer with bounding box-conditioned 2D center and depth regression for stable 3D translation recovery, and a 6D-aware bipartite matching cost.
- Through extensive experiments on the REAL275, CAMERA25, and HouseCat6D benchmarks, we demonstrate that YOPO significantly outperforms previous, more complex methods and sets a new state of the art in monocular category-level pose estimation.

II. RELATED WORK

A. Data Requirements for RGB 9D Object Pose Estimation

A closer inspection of existing research in monocular RGB category-level pose estimation reveals that few methods rely strictly on RGB images and their corresponding 9D pose annotations for both training and inference. Many state-of-the-art pipelines incorporate additional data and assumptions to achieve high performance.

A common requirement is the use of 3D CAD models of objects within the training categories, even if they are not required at inference time. These models are often used either to construct a canonical representation (e.g., NOCS [11]) [17], [19], [16] or to generate category-level shape priors that guide the learning process [15], [18]. Another prevalent dependency is the use of instance segmentation masks. Methods such as MSOS [17], DMSR [18], LaPose [16], MonoDiff9D [1], and DA-Pose [14] typically employ an off-the-shelf instance segmentation model (e.g., Mask R-CNN [24]) to isolate objects from the background. These masks are essential for cropping input images or feature maps to the object’s region of interest. Some methods incorporate pseudo-depth information by leveraging pre-trained monocular depth estimators (e.g., ZoeDepth [25] or DepthAnything [26]). This allows them to recover metric

TABLE I: Comparison of methods in terms of additional data requirements. ✓ denotes required, and ✗ denotes not required. We compare the RGB-only versions of *Synthesis* [20] and *CenterSnap* [30].

Method	CAD Model	Seg. Mask	Pseudo-depth
Synthesis (ECCV ‘20) [20]	✓	✓	✗
MSOS (RA-L ‘21) [17]	✓	✓	✗
CenterSnap (ICRA ‘22) [30]	✓	✗	✗
OLD-Net (ECCV ‘22) [15]	✓	✓	✓
FAP-Net (ICRA ‘24) [31]	✓	✓	✓
DMSR (ICRA ‘24) [18]	✓	✓	✓
LaPose (ECCV ‘24) [16]	✓	✓	✗
MonoDiff9D (ICRA ‘25) [1]	✗	✓	✓
DA-Pose (RA-L ‘25) [14]	✗	✓	✓
GIVEPose (CVPR ‘25) [19]	✓	✓	✗
YOPO (Ours)	✗	✗	✗

depth [17], [15], [14] or relative depth [18], effectively converting the monocular problem into a pseudo-RGB-D one to simplify 3D reasoning.

In contrast, our approach operates without any of these additional data dependencies. To the best of our knowledge, the only notable prior work with similarly minimal data assumptions is CenterPose [27]. However, the performance of this keypoint-based approach has been surpassed by the aforementioned more complex pipelines that leverage additional data [28], [29]. This highlights a gap in the literature for a method that can achieve state-of-the-art performance while adhering to a strict monocular RGB formulation. Our work aims to fill this gap, demonstrating that it is possible to surpass the performance of recent methods without resorting to external data sources such as CAD models, segmentation masks, or pseudo-depth maps (as shown in Table I and Table II).

B. Query-based Oriented Object Detection

Our research is motivated by the significant success of oriented object detection [32], [33] in the broader computer vision field. This 2D task aims to extract each object’s class c , 2D location $t \in \mathbb{R}^2$, in-plane rotation $R \in SO(2)$, and size $s \in \mathbb{R}_{>0}^2$ from a single RGB image. Recently, query-based architectures, pioneered by DETR [22], [23], have shown strong competitiveness by formulating the task as a direct set prediction problem [34], [35].

A key advantage of these query-based 2D detectors is their ability to operate in a clean, end-to-end fashion, requiring no additional data or priors beyond the input image and corresponding annotations. This contrasts with the complex pipelines commonly seen in 9D pose estimation. From this perspective, a central question motivating our work is whether the success of this streamlined, query-based paradigm can be transferred from 2D object detection to the more challenging domain of RGB, category-level 9D pose estimation. Notably, while such minimalist end-to-end detection approaches are more common in related domains like RGB-D [30], [36] or model-level [37], [38], [39], [40] pose estimation, they remain particularly scarce in the challenging RGB-only, category-level setting.

III. METHOD

A. Overall Architecture

Given an input RGB image $I \in \mathbb{R}^{H \times W \times 3}$ and the camera intrinsic matrix $K \in \mathbb{R}^{3 \times 3}$, our goal is to estimate, in a single forward pass, the category and 9D pose of all object instances:

$$\hat{y}_i = (c_i, \mathbf{R}_i, \mathbf{t}_i, \mathbf{s}_i), \quad i = 1, \dots, N, \quad (1)$$

where c_i denotes the object category, $\mathbf{R}_i \in \text{SO}(3)$ the 3D rotation, $\mathbf{t}_i \in \mathbb{R}^3$ the translation, and $\mathbf{s}_i \in \mathbb{R}^3$ the anisotropic scale.

Existing methods [18], [1], [19] typically decompose this task into two stages. In the first stage, they predict 2D object detections (c_i, \mathbf{b}_i) , where $\mathbf{b}_i \in \mathbb{R}^4$ denotes the 2D bounding box parameterized by (x, y, w, h) . In the second stage, pose estimation is performed on cropped image regions that contain individual object instances. To isolate objects from the background, these methods often employ separately trained instance segmentation models. Moreover, during training, collections of CAD models are commonly used to provide shape priors, enabling the pose estimation network to leverage existing geometric information. While effective, these dependencies on segmentation masks and CAD priors increase annotation costs and hinder generalization to novel object categories.

In contrast, our approach follows the design principles of DETR, which eliminate handcrafted priors. Our model directly predicts $(c_i, \mathbf{R}_i, \mathbf{t}_i, \mathbf{s}_i)$ from RGB images, end to end, using only object-category, pose, and size annotations. This design simplifies training and improves scalability across diverse object categories, without requiring instance masks or CAD models.

Specifically, our model builds on the transformer-based detector DINO [23], which extends DETR [22] with a two-stage refinement mechanism. As illustrated in Fig. 2a, the architecture comprises: (1) a multi-scale feature backbone for image feature extraction, (2) a transformer encoder that processes the feature maps, (3) a transformer decoder that refines object queries, and (4) task-specific prediction heads applied at both the proposal and refinement stages.

In the first stage, the transformer encoder processes the multi-scale backbone features, and a detection head predicts a set of reference points with associated objectness scores. The top- K proposals are selected based on these scores and are transformed into object queries \mathbf{q} , which carry both spatial (2D reference point) and semantic information. In the second stage, the transformer decoder refines these queries to produce \mathbf{p} , which are subsequently fed into parallel prediction heads for classification, 2D box regression, and 9D pose estimation.

B. Parallel Prediction Heads

At both the proposal and refinement stages, our architecture employs two parallel heads: a detection head and a pose-estimation head.

Detection Head. The detection head predicts object categories c_i and 2D bounding boxes \mathbf{b}_i . In the proposal stage, it generates coarse localization cues and selects the top- K object queries \mathbf{q} . During the refinement stage, it provides auxiliary supervision on categories and boxes; however, its predictions are not used at inference.

Although we optimize the detection head with a 2D bounding-box loss, these boxes require no manual annotation, as they can be automatically derived by projecting the annotated 3D cuboids onto the image plane (Table VI).

Pose Estimation Head. The pose-estimation head predicts the 9D pose parameters $(\mathbf{R}_i, \mathbf{t}_i, \mathbf{s}_i)$ via four regression branches: (1) 2D center offset, (2) depth, (3) rotation, and (4) scale. This head is supervised in both stages: during the proposal stage, auxiliary supervision encourages object queries to encode early geometric information; during the refinement stage, it outputs the final 9D pose estimates. The 3D translation \mathbf{t}_i is reconstructed by back-projecting the predicted 2D center and depth using the camera intrinsics K .

This parallel design allows the detection head to specialize in object localization, while the pose-estimation head focuses on 3D reasoning. Early supervision of the pose head ensures that object queries are geometry-aware even before refinement. By sharing the same set of object queries, our architecture jointly optimizes detection and pose estimation, enabling the two tasks to reinforce each other during training.

C. 2D Bounding Box-Conditioned 3D Prediction

For 3D translation, we employ a disentangled parameterization following Simonelli et al. [41], wherein the image-plane center and depth are predicted separately to enhance training stability in monocular settings. Specifically, the 2D center \mathbf{u}_i is normalized by the image width and height so that it lies in $[0, 1]^2$, and a sigmoid activation is applied to the corresponding outputs of the linear head. During inference, these normalized coordinates are rescaled by the original image dimensions to recover pixel-space positions. The final 3D translation \mathbf{t}_i is reconstructed by back-projecting the rescaled 2D center and the predicted depth using the camera intrinsic matrix K .

Bounding Box-Conditioned Center Prediction. Simonelli et al. [41] proposed regressing the 2D center \mathbf{u}_i as an offset from the center of the predicted 2D bounding box:

$$\mathbf{u}_i = \mathbf{b}_i^{\text{center}} + \Delta \mathbf{u}_i, \quad \Delta \mathbf{u}_i = \text{MLP}(\mathbf{p}_i), \quad (2)$$

where $\mathbf{p}_i \in \mathbb{R}^D$ is the refined embedding of the i -th object, obtained from the transformer decoder output, and $\mathbf{b}_i^{\text{center}}$ denotes the geometric center of the predicted bounding box \mathbf{b}_i . MLP refers to the linear head.

We further enhance the center-offset prediction head by explicitly conditioning it on the bounding-box parameters, as shown in Figure 2b. Specifically, instead of relying solely on the object query \mathbf{p}_i , we concatenate it with the predicted bounding box $\mathbf{b}_i \in \mathbb{R}^4$ to form an augmented input to the

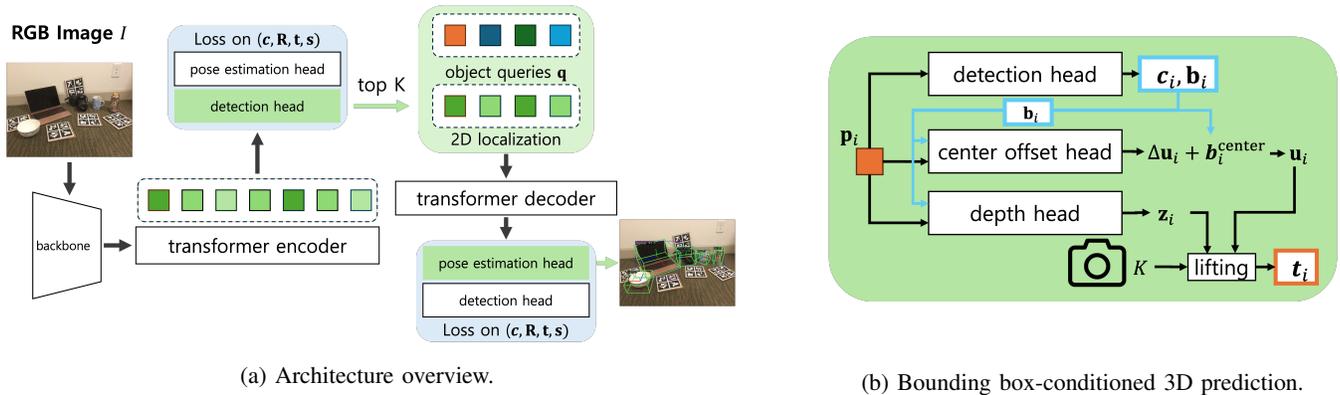


Fig. 2: Overview of our method. (a) The model predicts object properties from transformer-decoder outputs using task-specific heads. (b) The translation and depth head estimates 2D center locations as offsets from bounding-box centers, enabling 3D translation and depth recovery via back-projection. Predicted bounding boxes are concatenated with the input query to provide spatial information for accurate 2D center and depth estimation.

MLP:

$$\begin{aligned} \Delta \mathbf{u}_i &= \text{MLP}(\text{Concat}(\mathbf{p}_i, \mathbf{b}_i)), \quad \text{MLP} : \mathbb{R}^{D+4} \rightarrow \mathbb{R}^2, \\ \mathbf{u}_i &= \mathbf{b}_i^{\text{center}} + \Delta \mathbf{u}_i. \end{aligned} \quad (3)$$

This design allows the offset prediction to be informed not only by semantic features but also by explicit geometric guidance encoded in the bounding box.

At inference time, the predicted 2D center and depth are combined through perspective back-projection to yield the final 3D object translation \mathbf{t}_i in camera coordinates:

$$\mathbf{t}_i = z_i K^{-1} \begin{bmatrix} \mathbf{u}_i \\ 1 \end{bmatrix}. \quad (4)$$

This formulation maintains the structural benefits of the disentangled translation framework while enhancing its representational power through bounding box-conditioned center prediction.

Bounding Box-Conditioned Depth Prediction. Following the center head, we extend the depth predictor by concatenating the object query with the bounding-box parameters. This conditioning provides complementary geometric cues, enabling more stable depth regression and reducing scale ambiguities, ultimately improving translation accuracy.

D. 3D Matching Costs for Bipartite Matching

We follow DETR [22] and use one-to-one bipartite matching to assign predictions to ground-truth instances. The base matching cost includes classification, 2D bounding-box regression, and intersection-over-union (IoU) terms:

$$\mathcal{C}_{\text{match}} = \lambda_{\text{cls}}^c \cdot \mathcal{C}_{\text{cls}} + \lambda_{\text{bbox}}^c \cdot \mathcal{C}_{\text{bbox}} + \lambda_{\text{IoU}}^c \cdot \mathcal{C}_{\text{IoU}}, \quad (5)$$

where \mathcal{C}_{cls} is the classification cost, $\mathcal{C}_{\text{bbox}}$ is the ℓ_1 distance between the predicted and ground-truth 2D bounding boxes, \mathcal{C}_{IoU} is the negative IoU, and λ^c denotes the corresponding weights.

To handle 6D pose estimation, we add translation and rotation terms: $\mathcal{C}_{\text{trans}}$ is the Euclidean distance between 3D

translations, and \mathcal{C}_{rot} is the geodesic distance between rotation matrices, optionally accounting for object symmetries. The final matching cost is

$$\begin{aligned} \mathcal{C}_{\text{match}} &= \lambda_{\text{cls}}^c \cdot \mathcal{C}_{\text{cls}} + \lambda_{\text{bbox}}^c \cdot \mathcal{C}_{\text{bbox}} + \lambda_{\text{IoU}}^c \cdot \mathcal{C}_{\text{IoU}} \\ &\quad + \lambda_{\text{trans}}^c \cdot \mathcal{C}_{\text{trans}} + \lambda_{\text{rot}}^c \cdot \mathcal{C}_{\text{rot}}. \end{aligned} \quad (6)$$

We use the following default weights: $\lambda_{\text{cls}}^c=2.0$, $\lambda_{\text{bbox}}^c=5.0$, $\lambda_{\text{IoU}}^c=2.0$, $\lambda_{\text{trans}}^c=5.0$, $\lambda_{\text{rot}}^c=2.0$.

E. Implementation Details

Architecture. Our framework is built on top of the transformer-based DINO detector [23]. We largely follow DINO’s default settings, including the multi-scale backbone, encoder-decoder transformer structure, and a two-stage refinement with denoising training. Specifically, we adopt the same number of encoder and decoder layers, attention heads, hidden dimensions, and feed-forward network settings as DINO. Each task-specific component in the pose estimation head also follows the MLP structure of the detection head, except for the output dimension (or the input dimension when bounding-box information is concatenated with the object queries). Unlike DINO, we reduce the number of object queries to 100.

Pose Estimation Head. We represent rotations using the continuous 6D parameterization [42] and supervise them with a geodesic loss on $SO(3)$. Both the depth z and the anisotropic 3D scale $\mathbf{s} \in \mathbb{R}^3$ are regressed directly in linear space. Unless otherwise stated, we adopt class-wise rotation and scale prediction heads: the output dimensions are expanded by the number of categories, and the final predictions are selected according to the highest class confidence.

Losses and Training. We jointly optimize detection and pose using

$$\begin{aligned} \mathcal{L}_{\text{total}} &= \lambda_{\text{cls}} \mathcal{L}_{\text{cls}} + \lambda_{\text{bbox}} \mathcal{L}_{\text{bbox}} + \lambda_{\text{IoU}} \mathcal{L}_{\text{IoU}} \\ &\quad + \lambda_{\text{center2D}} \mathcal{L}_{\text{center2D}} + \lambda_{\text{depth}} \mathcal{L}_{\text{depth}} \\ &\quad + \lambda_{\text{rot}} \mathcal{L}_{\text{rot}} + \lambda_{\text{scale}} \mathcal{L}_{\text{scale}}. \end{aligned} \quad (7)$$

We use focal loss [43] for classification \mathcal{L}_{cls} , L1 for box \mathcal{L}_{bbox} and center prediction $\mathcal{L}_{center2D}$, GIoU [44] for IoU loss \mathcal{L}_{IoU} , and L2 for depth \mathcal{L}_{depth} and scale \mathcal{L}_{scale} . We use AdamW [45] with learning rate 1×10^{-4} for batch size 16 on $4 \times A6000$ GPUs. All models are initialized from DINO pretrained on COCO to leverage strong object detection performance.

IV. EXPERIMENTS

A. Datasets and Metrics

Datasets. We evaluate YOPO on three widely used benchmarks for category-level 9D pose estimation: CAMERA25, REAL275 [11], and HouseCat6D [46]. CAMERA25 is a synthetic dataset containing 275K training images and 25K test images across six object categories. REAL275 is a real-world dataset with the same categories as CAMERA25. It consists of 4.3K training images from 7 scenes and 2.75K test images from 6 scenes, with three unseen object instances per category in the test split. HouseCat6D features 194 high-fidelity 3D object models across 10 household categories, with 20K training, 1.4K validation, and 3K test images captured in 41 real scenes.

Evaluation Metrics. We follow standard protocols [11], [12] and report two main metrics:

- **3D IoU:** We report mean Average Precision (mAP) at 3D bounding-box IoU thresholds of 50% and 75%. This metric jointly reflects the accuracy of pose and 3D scale estimation.
- n° m **cm:** We also report the mAP of predictions in which the rotation error is below n° and the translation error is under m cm. This directly evaluates geometric accuracy and is commonly used for 6D pose estimation.

Following DMSR [18], we adopt 50% and 75% as the thresholds for 3D IoU evaluation, and we report pose accuracy under the 10° , 10 cm, and 10° -10 cm criteria for assessing rotation and translation errors.

Experimental Settings. Unless otherwise specified, we train YOPO for 12 epochs on the combined CAMERA25 and REAL275 datasets using 9D pose annotations and 2D bounding boxes, and evaluate using a single model. YOPO* is obtained by fine-tuning YOPO on REAL275 for 12 epochs. For HouseCat6D, we adopt the same training schedule and hyperparameters, but set the learning rate to 2×10^{-4} , which is twice that used for CAMERA25 and REAL275. We apply random pixel translations and horizontal flips for data augmentation.

B. Comparison with Previous Methods

Table II compares YOPO with existing state-of-the-art methods on the CAMERA25 and REAL275 datasets. YOPO consistently outperforms all prior RGB-only approaches on both datasets. On CAMERA25, YOPO with a Swin-L backbone [47] achieves **46.6%** IoU₅₀, **11.8%** IoU₇₅, and **38.7%** under the 10° 10cm criterion. On REAL275, YOPO achieves **71.6%** IoU₅₀ and **52.8%** 10° 10cm, outperforming prior RGB-only methods across all metrics. With additional fine-tuning on the REAL275 training split (denoted as YOPO*),

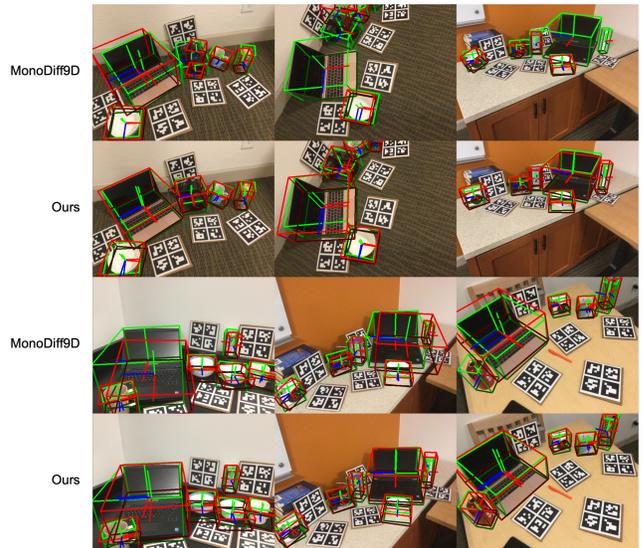


Fig. 3: Qualitative comparison of pose estimation results on the REAL275 dataset. We compare our model with MonoDiff9D [1]. Predicted poses are shown in red, while ground-truth annotations are shown in green.

our method further improves to **79.6%** IoU₅₀ and **54.1%** 10° 10cm. YOPO with the lighter ResNet-50 backbone [48] also maintains this overall superiority.

Figure 3 presents qualitative results comparing our method with the previous state-of-the-art MonoDiff9D [1] on the REAL275 dataset. The top two rows illustrate the advantages of our end-to-end framework for joint object detection and pose estimation. Unlike MonoDiff9D, which relies on a separately trained instance segmentation model and suffers from missed detections and false positives, YOPO directly detects objects and estimates their poses in a unified pipeline, thereby reducing error propagation. The bottom two rows demonstrate the accuracy of our method in estimating 3D translation, rotation, and scale. YOPO’s predictions consistently align more closely with the ground-truth cuboids than those of MonoDiff9D, especially in cluttered scenes with varying object scales.

C. Ablation Study and Discussion

We conduct comprehensive ablation studies on the REAL275 dataset to assess the contribution of each key design component of YOPO. Unless otherwise noted, all ablations use direct 2D bounding-box supervision rather than projected cuboids.

a) *Component-wise Ablation:* We ablate YOPO on REAL275 with emphasis on IoU₅₀ and the 10° 10cm metric. Starting from a model with all components disabled, performance is modest (51.2 IoU₅₀ / 26.1 10° 10cm). Introducing box-conditioning on the center head (**BC {center}**) yields a sizable overlap gain (59.8 IoU₅₀; +8.6), albeit at the cost of geometric accuracy (21.8 10° 10cm). Adding data augmentation substantially recovers the latter (34.1 10° 10cm) while keeping overlap stable (59.1 IoU₅₀). With augmentation but without class-wise heads, proper weight scaling (**WS**) within 3D-aware matching raises performance to 66.6 IoU₅₀ and

TABLE II: Comparison on the CAMERA25 and REAL275 datasets. For each result, **bold** is used to indicate the top-performing method among all methodologies. An underline highlights the best performing method *when our proposed method is excluded*.

Method	CAMERA25					REAL275				
	IoU ₅₀	IoU ₇₅	10cm	10°	10°10cm	IoU ₅₀	IoU ₇₅	10cm	10°	10°10cm
Synthesis (ECCV 2020) [20]	-	-	-	-	-	-	-	34.0	14.2	4.8
MSOS (RA-L 2021) [17]	32.4	5.1	29.7	60.8	19.2	23.4	3.0	39.5	29.2	9.6
CenterSnap-RGB (ICRA 2022) [30]	-	-	-	-	-	31.5	-	-	-	30.1
OLD-Net (ECCV 2022) [15]	32.1	5.4	30.1	74.0	23.4	25.4	1.9	38.9	37.0	9.8
FAP-Net (ICRA 2024) [31]	39.2	6.7	36.0	80.4	<u>29.8</u>	<u>36.8</u>	5.2	<u>49.7</u>	49.6	24.5
DMSR (ICRA 2024) [18]	34.6	6.5	32.3	<u>81.4</u>	27.4	28.3	6.1	37.3	<u>59.5</u>	23.6
LaPose (ECCV 2024) [16]	-	-	-	-	-	17.5	2.6	44.4	-	30.5
MonoDiff9D (ICRA 2025) [1]	35.2	<u>6.7</u>	33.6	80.1	28.2	31.5	<u>6.3</u>	41.0	56.3	25.7
DA-Pose (RA-L 2025) [14]	<u>41.4</u>	6.1	<u>40.5</u>	60.8	24.6	28.1	3.6	45.8	27.5	13.4
GIVEPose (CVPR 2025) [19]	-	-	-	-	-	20.1	-	45.9	-	<u>34.2</u>
YOPO R50 (Ours)	41.4	7.9	36.3	78.2	30.1	67.1	16.6	75.6	54.0	40.7
YOPO Swin-L (Ours)	46.6	11.8	43.1	88.7	38.7	71.6	16.4	77.8	69.6	52.8
YOPO Swin-L* (Ours)	-	-	-	-	-	79.6	19.6	84.4	66.0	54.1

TABLE III: Ablation on **REAL275**. ✓: component enabled, ✗: disabled. The shaded rows indicate the selection for our final model, corresponding to the main table (Table II).

	BC	Aug	CW	WS	RFT	IoU ₅₀	IoU ₇₅	10cm	10°	10°10cm
						IoU ₅₀	IoU ₇₅	10cm	10°	10°10cm
R50	✗	✗	✗	✗	✗	51.2	10.9	58.0	46.8	26.1
	center	✗	✗	✗	✗	59.8	10.9	65.7	33.3	21.8
	center	✓	✗	✗	✗	59.1	11.0	66.2	55.7	34.1
	center	✓	✓	✗	✗	55.7	10.6	59.7	59.2	33.5
	center	✓	✗	✓	✗	66.6	11.8	76.7	54.9	42.4
	center	✓	✓	✓	✗	64.2	16.7	70.0	54.1	37.4
	center, z	✓	✓	✓	✗	67.1	16.6	75.6	54.0	40.7
	center, z	✓	✓	✓	✓	71.1	17.7	77.6	60.5	46.6
	Swin-L	center, z	✓	✓	✓	✗	71.6	16.4	77.8	69.6
center, z		✓	✓	✓	✓	79.6	19.6	84.4	66.0	54.1

TABLE IV: Ablation on **REAL275** with respect to 3D-aware matching cost and weights for the loss.

BC	MC	λ_{rot}	λ_{depth}	λ_{scale}	IoU ₅₀	IoU ₇₅	10cm	10°	10°10cm
center	✗	5.0	5.0	5.0	55.7	10.6	59.7	<u>59.2</u>	33.5
center	✓	5.0	5.0	5.0	55.8	8.9	64.0	56.7	<u>37.8</u>
center	✓	5.0	50.0	50.0	67.3	<u>15.1</u>	71.9	50.0	35.0
center	✓	5.0	50.0	5.0	62.2	13.7	67.6	60.5	40.5
center	✓	5.0	50.0	50.0	<u>64.2</u>	16.7	<u>70.0</u>	54.1	37.4

42.4 10°10cm. Extending BC to also condition the depth head (**BC {center, z}**) provides a consistent overlap gain (67.1 IoU₅₀) with comparable 10°10cm (40.7). Further gains come from fine-tuning on REAL275 (**RFT**), which raises the ResNet-50 model to 71.1 on IoU₅₀ and 46.6 on 10°10cm; with a stronger Swin-L backbone, the same recipe reaches 79.6 on IoU₅₀ and 54.1 on 10°10cm. Overall, most of the performance improvement is driven by 3D-aware matching, box-conditioning, and supporting components. The shaded row indicates our final model, which offers a balanced and strong performance.

b) *Effect of 3D-aware Matching Costs and Loss-Weight Scaling*: Table IV shows that 3D-aware matching costs mostly improve pose accuracy, but their impact is amplified when combined with proper loss-weight scaling. Without 3D-aware matching costs and with uniform loss weights, the

TABLE V: Ablation study on bounding box-conditioned prediction.

Center	Rotation	Size	Z	IoU ₅₀	IoU ₇₅	10cm	10°	10°10cm
✗	✗	✗	✗	66.0	12.5	74.4	52.1	39.5
✓	✗	✗	✗	64.2	16.7	70.0	54.1	37.4
✓	✓	✗	✗	64.3	14.0	74.8	47.7	36.3
✓	✗	✓	✗	65.0	11.9	75.0	52.1	39.2
✓	✗	✗	✓	67.1	<u>16.6</u>	75.6	<u>54.0</u>	<u>40.7</u>
✓	✓	✗	✓	70.9	13.8	79.0	48.4	39.1
✓	✗	✓	✓	65.4	14.2	73.6	49.2	36.0
✓	✓	✓	✓	<u>67.6</u>	12.6	<u>75.8</u>	59.6	47.2

TABLE VI: Comparison of direct 2D bounding box supervision (✓) vs. projected 3D cuboid supervision (✗) on **REAL275**. All other settings are held constant.

Backbone	BC	Box	IoU ₅₀	IoU ₇₅	10cm	10°	10°10cm
R50	center	✓	64.2	16.7	70.0	54.1	37.4
	center	✗	62.1	15.2	67.2	57.4	38.7
Swin-L	center	✓	69.3	14.7	76.9	65.6	49.3
	center	✗	67.1	15.8	78.0	67.9	53.3

model achieves 55.7 on IoU₅₀ and 33.5 on 10°10cm. Enabling 3D-aware matching costs at the same weights leaves the overlap essentially unchanged (55.8 IoU₅₀) but improves the 10°10cm metric to 37.8 (+4.3). Reweighting λ_{depth} and λ_{scale} to 50 without matching costs increases the overlap to 67.3 IoU₅₀ (+11.6) while yielding a modest 10°10cm of 35.0. Combining 3D-aware matching with targeted reweighting produces the strongest performance: specifically, setting $\lambda_{rot} = 5$, $\lambda_{depth} = 50$, and $\lambda_{scale} = 5$ achieves 37.4 on 10°10cm and 64.2 on IoU₅₀. Consequently, introducing 3D-aware matching costs together with appropriate loss-weight scaling substantially improves performance. This loss-weight configuration was adopted for our final model.

c) *Effect of Bounding Box-Conditioned Prediction*: Table V isolates the effect of conditioning different prediction heads on the 2D box. Without any conditioning, the model

TABLE VII: Comparison with state-of-the-art RGB and RGB-D methods on REAL275 and HouseCat6D.

		REAL275				
Method		IoU ₅₀	IoU ₇₅	5° 5 cm	10° 5 cm	10° 10 cm
RGB-D	NOCS [11]	78.0	30.1	10.0	25.2	-
	GPV-Pose [49]	-	64.4	42.9	73.3	-
	AG-Pose [50]	83.7	79.5	61.7	83.1	-
	SpotPose [51]	84.1	81.2	64.8	88.2	-
RGB	MonoDiff9D [1]	31.5	6.3	4.4	9.6	25.7
	GIVEPose [19]	20.1	-	-	-	34.2
	YOPO (Ours)	79.6	19.6	5.9	26.7	54.1

		HouseCat6D				
Method		IoU ₂₅	IoU ₅₀	5° 5 cm	10° 5 cm	10° 10 cm
RGB-D	NOCS [11]	50.0	21.2	-	-	-
	GPV-Pose [49]	74.9	50.7	4.6	22.7	-
	AG-Pose [50]	81.8	62.5	12.0	35.8	-
	SpotPose [51]	89.1	77.0	24.5	54.8	-
RGB	YOPO (Ours)	71.3	34.8	5.3	22.1	33.3

attains 66.0 on IoU₅₀ and 39.5 on 10°10cm. Conditioning only the center slightly degrades the main metrics (64.2 on IoU₅₀ / 37.4 on 10°10cm), and adding conditioning to rotation or size (on top of the center) yields limited or inconsistent gains. In contrast, conditioning depth together with the center produces a clear joint improvement: 67.1 on IoU₅₀ and 40.7 on 10°10cm. While extending conditioning further can improve a single metric, it comes with trade-offs on the complementary objective. Therefore, we adopt conditioning only the center and depth heads as the default, as this choice offers the best balance between overlap and geometric accuracy.

d) Is exact 2D box supervision necessary?: In Table VI, we examine whether our method requires precise 2D bounding boxes during training. Specifically, we compare the standard setup that uses directly annotated 2D boxes (✓) with a weaker alternative in which 2D boxes are derived from projected 3D cuboids (✗), which are less accurate. On ResNet-50, precise 2D boxes yield slightly better 3D IoU (+1.2 points on IoU₇₅) and translation accuracy (+2.8 points on the 10 cm metric), whereas the projected-cuboid boxes still produce competitive results and even achieve higher rotation accuracy (+3.3 points on the 10° metric). With the Swin-L backbone, both strategies deliver strong performance, indicating that our method remains robust even under weaker box supervision. Overall, YOPO does not strictly rely on highly accurate 2D bounding boxes, which can further reduce annotation costs in practical settings.

e) Comparison with RGB-D and RGB Methods: Table VII compares YOPO (Swin-L) against RGB-D and RGB methods on REAL275 and HouseCat6D. On REAL275, YOPO decisively surpasses RGB-only baselines in both overlap and geometric accuracy, approaching RGB-D performance in overlap while trailing on stricter criteria (e.g., IoU₇₅ and 10°5cm). On HouseCat6D, YOPO achieves 34.8 on IoU₅₀ and 5.3 on 5°5cm, delivering performance comparable to GPV-Pose on the latter (4.6) and exceeding NOCS in IoU₅₀ (21.2), though it remains behind the strongest RGB-D system (SpotPose). Notably, methods reported on HouseCat6D leverage ground-truth segmentation masks, whereas YOPO operates purely on RGB images without such aux-

iliary supervision. Overall, YOPO markedly narrows the gap between RGB and RGB-D methods and establishes a clear lead among RGB-only approaches, underscoring the effectiveness of our end-to-end framework for category-level 9D pose estimation from RGB.

f) Inference Time: Our model performs joint object detection and 9D pose estimation for multiple objects in a single RGB image in a single forward pass. It achieves 21.3 frames per second (FPS) with ResNet-50 and 7.7 FPS with Swin-Large on an NVIDIA A6000 GPU, combining state-of-the-art accuracy with efficient end-to-end inference.

V. CONCLUSION

We introduced YOPO, a single-stage, transformer-based framework for monocular, category-level 9D pose estimation that operates truly end to end—without CAD models, shape priors, pseudo-depth, or instance-mask supervision. Built on DINO, YOPO adds a parallel pose head and a bounding-box-conditioned 3D module, enabling strong direct estimation of rotation, translation, and anisotropic scale from RGB alone in a single forward pass. Across standard benchmarks, including REAL275 and HouseCat6D, YOPO establishes a new state of the art among RGB-only methods and substantially narrows the gap to RGB-D systems, while maintaining a cost-effective and scalable design suitable for real-world deployment. Looking ahead, we see YOPO as a simple, strong baseline for RGB-only 9D perception, and an extensible platform for exploring robustness to occlusion and domain shift, broader category coverage, and the integration of temporal cues.

ACKNOWLEDGEMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2025-02653113, High-Performance Research AI Computing Infrastructure Support at the 2 PFLOPS Scale). This research was supported by Deep-Tech Tips funded by Ministry of SMEs and Startups.

REFERENCES

- [1] J. Liu, W. Sun, H. Yang, J. Zheng, Z. Geng, H. Rahmani, and A. Mian, “Monodiff9d: Monocular category-level 9d object pose estimation via diffusion model,” in *ICRA*, 2025.
- [2] B. Fu, S. K. Leong, X. Lian, and X. Ji, “6d robotic assembly based on rgb-only object pose estimation,” in *IROS*. IEEE, 2022, pp. 4736–4742.
- [3] J. Liu, W. Sun, H. Yang, C. Liu, X. Zhang, and A. Mian, “Domain-generalized robotic picking via contrastive learning-based 6-d pose estimation,” *IEEE Transactions on Industrial Informatics*, vol. 20, no. 6, pp. 8650–8661, 2024.
- [4] F. Tang, Y. Wu, X. Hou, and H. Ling, “3d mapping and 6d pose computation for real time augmented reality on cylindrical objects,” *IEEE TCSVT*, vol. 30, no. 9, pp. 2887–2899, 2019.
- [5] E. Marchand, H. Uchiyama, and F. Spindler, “Pose estimation for augmented reality: a hands-on survey,” *IEEE TVCG*, vol. 22, no. 12, pp. 2633–2651, 2015.
- [6] Z. Yuan, X. Song, L. Bai, Z. Wang, and W. Ouyang, “Temporal-channel transformer for 3d lidar-based video object detection for autonomous driving,” *IEEE TCSVT*, vol. 32, no. 4, pp. 2068–2078, 2021.

- [7] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *CVPR*, 2017, pp. 1907–1915.
- [8] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," in *CVPR*, 2019, pp. 4561–4570.
- [9] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *CVPR*, 2019, pp. 3343–3352.
- [10] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation," in *CVPR*, 2020, pp. 11 632–11 641.
- [11] 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 *CVPR*, 2019, pp. 2642–2651.
- [12] W. Chen, X. Jia, H. J. Chang, J. Duan, L. Shen, and A. Leonardis, "Fs-net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism," in *CVPR*, 2021, pp. 1581–1590.
- [13] J. Liu, W. Sun, C. Liu, H. Yang, X. Zhang, and A. Mian, "Mh6d: Multi-hypothesis consistency learning for category-level 6-d object pose estimation," *IEEE TNNLS*, 2024.
- [14] H. Yang, W. Sun, J. Liu, J. Zheng, Z. Zeng, and A. Mian, "Rgb-based category-level object pose estimation via depth recovery and adaptive refinement," *IEEE RA-L*, 2025.
- [15] 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 *ECCV*. Springer, 2022, pp. 220–236.
- [16] R. Zhang, Z. Huang, G. Wang, C. Zhang, Y. Di, X. Zuo, J. Tang, and X. Ji, "Lapose: Laplacian mixture shape modeling for rgb-based category-level object pose estimation," in *ECCV*. Springer, 2024, pp. 467–484.
- [17] T. Lee, B.-U. Lee, M. Kim, and I. S. Kweon, "Category-level metric scale object shape and pose estimation," *IEEE RA-L*, vol. 6, no. 4, pp. 8575–8582, 2021.
- [18] 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 *ICRA*. IEEE, 2024, pp. 2036–2042.
- [19] Z. Huang, G. Wang, C. Zhang, R. Zhang, X. Li, and X. Ji, "Givepose: Gradual intra-class variation elimination for rgb-based category-level object pose estimation," in *CVPR*, 2025, pp. 22 055–22 066.
- [20] X. Chen, Z. Dong, J. Song, A. Geiger, and O. Hilliges, "Category level object pose estimation via neural analysis-by-synthesis," in *ECCV*. Springer, 2020, pp. 139–156.
- [21] K. Chen and Q. Dou, "Sgpa: Structure-guided prior adaptation for category-level 6d object pose estimation," in *ICCV*, 2021, pp. 2773–2782.
- [22] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*. Springer, 2020, pp. 213–229.
- [23] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. Ni, and H.-Y. Shum, "DINO: DETR with improved denoising anchor boxes for end-to-end object detection," in *ICLR*, 2023. [Online]. Available: <https://openreview.net/forum?id=3mRwyG5one>
- [24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2961–2969.
- [25] S. F. Bhat, R. Birkel, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," *arXiv preprint arXiv:2302.12288*, 2023.
- [26] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *NeurIPS*, vol. 37, pp. 21 875–21 911, 2024.
- [27] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield, "Single-stage keypoint-based category-level object pose estimation from an rgb image," in *ICRA*. IEEE, 2022, pp. 1547–1553.
- [28] S. Yu, D.-H. Zhai, Y. Xia, D. Li, and S. Zhao, "Cattrack: Single-stage category-level 6d object pose tracking via convolution and vision transformer," *IEEE Transactions on Multimedia*, vol. 26, pp. 1665–1680, 2023.
- [29] Y. Mei, S. Wang, Z. Li, J. Sun, and G. Wang, "Multi-modal 6-dof object pose tracking: integrating spatial cues with monocular rgb imagery," *International Journal of Machine Learning and Cybernetics*, vol. 16, no. 2, pp. 1327–1340, 2025.
- [30] M. Z. Irshad, T. Kollar, M. Laskey, K. Stone, and Z. Kira, "Centersnap: Single-shot multi-object 3d shape reconstruction and categorical 6d pose and size estimation," in *ICRA*. IEEE, 2022, pp. 10 632–10 640.
- [31] 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 *ICRA*. IEEE, 2024, pp. 2043–2050.
- [32] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented r-cnn for object detection," in *ICCV*, 2021, pp. 3520–3529.
- [33] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning roi transformer for oriented object detection in aerial images," in *CVPR*, 2019, pp. 2849–2858.
- [34] Y. Zeng, Y. Chen, X. Yang, Q. Li, and J. Yan, "Ars-detr: Aspect ratio-sensitive detection transformer for aerial oriented object detection," *IEEE TGRS*, vol. 62, pp. 1–15, 2024.
- [35] H. Lee, M. Song, J. Koo, and J. Seo, "Hausdorff distance matching with adaptive query denoising for rotated detection transformer," in *WACV*. IEEE, 2025, pp. 1872–1882.
- [36] M. Z. Irshad, S. Zakharov, R. Ambrus, T. Kollar, Z. Kira, and A. Gaidon, "Shapo: Implicit representations for multi-object shape, appearance, and pose optimization," in *ECCV*. Springer, 2022, pp. 275–292.
- [37] D. Maji, S. Nagori, M. Mathew, and D. Poddar, "Yolo-6d-pose: Enhancing yolo for single-stage monocular multi-object 6d pose estimation," in *3DV*. IEEE, 2024, pp. 1616–1625.
- [38] 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 *CoRL*. PMLR, 2023, pp. 1060–1070.
- [39] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *CVPR*, 2018, pp. 292–301.
- [40] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *RSS*, 2018.
- [41] A. Simonelli, S. R. Buló, L. Porzi, M. López-Antequera, and P. Kotschieder, "Disentangling monocular 3d object detection," in *ICCV*, 2019, pp. 1991–1999.
- [42] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *CVPR*, 2019, pp. 5745–5753.
- [43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017, pp. 2980–2988.
- [44] H. Rezaatofghi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *CVPR*, 2019, pp. 658–666.
- [45] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019. [Online]. Available: <https://openreview.net/forum?id=Bkg6RiCqY7>
- [46] H. Jung, S.-C. Wu, P. Ruhkamp, G. Zhai, H. Schieber, G. Rizzoli, P. Wang, H. Zhao, L. Garattoni, S. Meier *et al.*, "Housecat6d-a large-scale multi-modal category level 6d object perception dataset with household objects in realistic scenarios," in *CVPR*, 2024, pp. 22 498–22 508.
- [47] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021, pp. 10 012–10 022.
- [48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
- [49] Y. Di, R. Zhang, Z. Lou, F. Manhardt, X. Ji, N. Navab, and F. Tombari, "Gpv-pose: Category-level object pose estimation via geometry-guided point-wise voting," in *CVPR*, 2022, pp. 6781–6791.
- [50] X. Lin, W. Yang, Y. Gao, and T. Zhang, "Instance-adaptive and geometric-aware keypoint learning for category-level 6d object pose estimation," in *CVPR*, 2024, pp. 21 040–21 049.
- [51] H. Ren, W. Yang, S. Zhang, and T. Zhang, "Rethinking correspondence-based category-level object pose estimation," in *CVPR*, 2025, pp. 1170–1179.