Category-Level and Open-Set Object Pose Estimation for Robotics

Peter Hönig, Matthias Hirschmanner, and Markus Vincze

***Abstract*— Object pose estimation enables a variety of tasks in computer vision and robotics, including scene understanding and robotic grasping. The complexity of a pose estimation task depends on the unknown variables related to the target object. While instance-level methods already excel for opaque and Lambertian objects, category-level and open-set methods, where texture, shape, and size are partially or entirely unknown, still struggle with these basic material properties. Since texture is unknown in these scenarios, it cannot be used for disambiguating object symmetries, another core challenge of 6D object pose estimation. The complexity of estimating 6D poses with such a manifold of unknowns led to various datasets, accuracy metrics, and algorithmic solutions. This paper compares datasets, accuracy metrics, and algorithms for solving 6D pose estimation on the category-level. Based on this comparison, we analyze how to bridge category-level and open-set object pose estimation to reach generalization and provide actionable recommendations.**

***Index Terms*— object pose estimation, symmetry handling, instance level, category-level, novel object, open set**

I. INTRODUCTION

Object pose estimation is necessary in robotics for tasks such as robotic grasping [1]. If the geometry of a target object is known, its 6D pose, defined by the rotation $\mathbf{R}$ and translation $\mathbf{t}$, is sufficient to locate it in a $SE(3)$ space. This definition is insufficient as soon as the target object geometry is only partially known, as in the case of category-level object pose estimation. Open-set object pose estimation is even more complex than category-level object pose estimation since both geometry and texture are entirely unknown. In category-level and open-set object pose estimation, a sole 6D pose leaves unknowns to describe the object for tasks such as grasping. In these cases, additional information is necessary, such as a 7D pose ($\mathbf{R}$, $\mathbf{t}$, and s for scale), a 9D pose ($\mathbf{R}$, $\mathbf{t}$, and $\mathbf{s}$, a 3D vector with x, y, z dimensions of the aligned bounding box), or a 6D pose in combination with a shape reconstruction. The differences between instance-level, category-level, and open-set object pose estimation are illustrated in Fig. 1. The three circles in Fig. 1 represent the prior knowledge available to a pose estimation algorithm during a training or onboarding stage. During the inference stage existing category-level algorithms [2], [3], [4] do not require a 3D object model. However, in open-set object pose estimation, recent methods do require an object model during inference [5], or they reconstruct an object mesh from multiview RGB images during an onbording stage [6]. These reconstructions however are prone to reconstruction noise and the full object surface needs to be visible in order for a full reconstruction. This does not allow one-shot pose estimation, where only a single frame is available.

All authors are with the Automation and Control Institute, Faculty of Electrical Engineering, TU Wien, 1040 Vienna, Austria {hoenig, hirschmanner, vincze}@acin.ac.tuwien.at

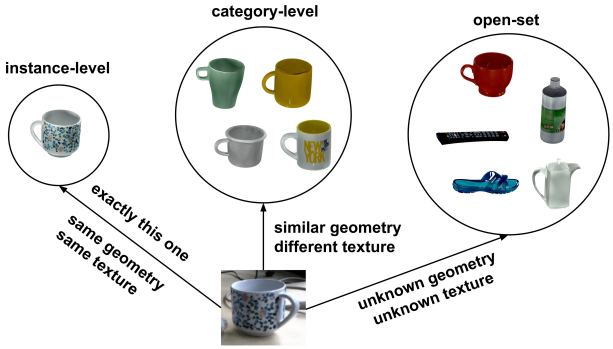

Fig. 1. **Comparison of instance-level, category-level, and open-set object pose estimation.** The complexity of the pose estimation task is increasing from left (instance-level) to right (open-set) due to the number of unknowns.

In order to make 3D object models during inference obsolete, knowledge has to be induced during a training stage. Consequently, canonicalization is required. Canonicalization describes the centering and alignment of objects canonically in the $SE(3)$ space. To address the issues of texture and geometry variation, a color encoding of object geometry is proposed in [2], namely the Normalized Object Coordinate Space (NOCS). NOCS describes a dimensionless $1 \times 1 \times 1$ cube, where the x, y, and z coordinates of canonically oriented objects are mapped to RGB values. This geometrical color encoding is used in other category-level and open-set object pose estimation solutions [7], [8], [9], [10], [11], [3], [12], becoming the de-facto standard intermediate representation in the field.

Besides dealing with texture and shape variations, properly disambiguating object symmetries remains challenging in category-level object pose estimation. Not accounting for object symmetries may prohibit optimization algorithms from converging correctly. While NOCS does disambiguate geometrically symmetric objects with distinct textures, it does not account for textureless objects and pointcloud-only input data. While instance-level object pose estimation uses distinct textures to disambiguate geometric symmetries, category-level object pose estimation deals with variation in object texture. This texture variety is handled differently during training of neural networks for category-level object pose estimation [11], [3].

This comparative study analyzes category-level object pose estimation papers [2], [7], [8], [13], [9], [10], [4], [14], [11], [3], [15], [12] focusing on symmetry handling and how

https://doi.org/10.34749/3061-0710.2025.13

Creative Commons Attribution 4.0 International License

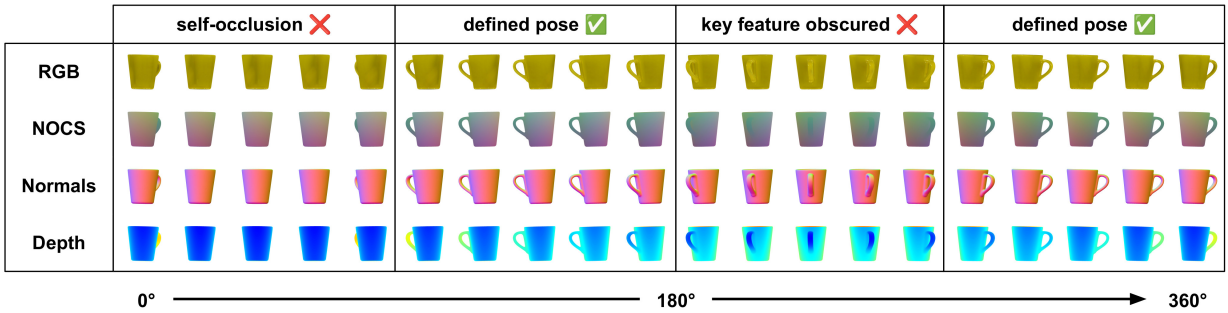

Fig. 2. **Comparing Input Modalities.** A mug is rotated 360° around the y-axis to showcase how a key feature (the handle) is self-occluded in a 90° range, clearly exposed in a 180° range, and obscured due to the uniform texture in a 90° range. The various input modalities are shown to highlight the mug handle stronger (depth, normals) or weaker (RGB, NOCS).

to bridge category-level and open-set object pose estimation. We limit our investigation to algorithms predicting the object pose from a single frame, with no 3D object model available during inference. We only consider algorithms which are evaluated either on the CAMERA [2] or REAL275 [2] dataset. Algorithms that use stereo vision [16] or consecutive frames [17] or are solely evaluated on other datasets are not included in this study. While previous works summarize and review the state of the art in object pose estimation in general [1], [18], [19], we explicitly focus on category-level object pose estimation and how to bridge this technique to the open-set domain, without requiring 3D object models during the inference step. Our contributions can be summarized as follows:

- A concise review of the state of the art in category-level object pose estimation with an emphasis on input modalities, network architectures, 6D pose solvers, and rotational symmetry handling.
- Actionable recommendations for tailoring future category-level object pose estimation methods for generalization beyond known categories to bridge to the open-set domain.

The following section starts by discussing types of input modalities. We will discuss how input modalities influence symmetry disambiguation. The subsequent section reviews network architectures. In this section we will discuss how network architectures can handle symmetric objects differently and how network types relate to model performances. Next, we discuss types symmetry handling, their advantages and limitations. We continue with a section about 6D pose solvers, discussing deterministic and learned variants and we elaborate on pose estimation performance metrics used for comparison. We conclude with experiments and results, discussing current performances of category-level object pose estimation algorithms and give actionable recommendations for potential improvements based on the findings during our literature research. The papers are compared in Table I.

II. INPUT MODALITIES

The choice of input modalities influences how symmetries are resolved [1]. For specific object geometries, RGB-only leads to ambiguous views, where key features are not clearly exposed, e.g., when the handle of a textureless mug directly points toward the camera. This phenomenon is depicted in Fig. 2. Depth input provides complementary geometric cues, which resolves such ambiguities. Furthermore, since texture and shape are varying in category-level object pose estimation, depth or normals input provide actual geometric surface information. Since category-level object pose estimation algorithms do not access 3D object meshes during the inference stage, algorithms using depth or normals are in favor. In robotic applications, depth data is abundant [1], [20] making algorithms that use depth modality the preferred option.

The papers [2], [7], [8], [9], [3], [15], [12] use RGB input for their algorithms, as listed in Table I. [2], [7] use RGB for establishing 2D-3D correspondences and depth for transforming normalized 3D to metric 3D coordinates. [9], [15] use RGB and depth for predicting 2D-3D correspondences and depth again for 6D pose solving between normalized and metric 3D. [13], [10], [4], [14], [11] rely on depth only for pose estimation and use RGB for the object detection stage to estimate region proposals. [8], [12] use solely RGB data, while [12] extracts DINOv2 features from RGB before processing further. [3] use RGB, depth, and DINO features.

III. NETWORK ARCHITECTURES

Network architectures vary between the selected papers. As Table I indicates, recent advancements in computer vision research are reflected in the development of novel category-level object pose estimation methods. While traditional CNN architectures dominated from 2019 to 2021, Graph Convolutional Networks (GCNs) [25] were adopted in 2022. [3] use diffusion in 2024, and [11], [12] use transformer models in 2023 and 2024. When comparing pose estimation accuracy in the subsequent chapters, one must consider that GCNs, transformers, and diffusion models need extended training and inference times compared to CNNs due to their different network characteristics. This is due to transformers processing the input in multiple heads [26], diffusion models requiring multiple denoising steps [27] and GCNs dealing with not only 2D but 3D data.

https://doi.org/10.34749/3061-0710.2025.13

Creative Commons Attribution 4.0 International License

TABLE I

COMPARISON OF CATEGORY-LEVEL OBJECT POSE ESTIMATION PAPERS. PAPERS PRESENTING ALGORITHMS PREDICTING CATEGORY-LEVEL OBJECT POSE FROM SINGLE FRAMES.

Year	Paper	Input		NOCS	Symmetry Handling	Network	6D Pose Solver
		RGB	Depth				
2019	Wang et al. [2]	✓	✓	✓	sym. transform loss [21]	Mask R-CNN like	Umeyama [22]
2020	Tian et al. [7]	✓	✓	✓	sym. transform loss [21]	Encoder-decoder CNN	Umeyama [22]
2021	Lee et al. [8]	✓	×	✓	none	Encoder-decoder CNN	Umeyama [22]
2021	Chen et al. [13]	×	✓	×	sym. transform loss [21]	Encoder-decoder CNN	Direct regression
2021	Wang et al. [9]	✓	✓	✓	none	Recurrent reconstruction CNN	Umeyama [22]
2022	Zhang et al. [10]	×	✓	✓	sym. transform loss [21]	3D GCN	Direct regression
2023	Wan et al. [4]	×	✓	(✓)	none	3D GCN	Anisotropic scaling
2023	Wang et al. [14]	×	✓	✓	none	CNNs + MLPs	Direct regression
2023	Zou et al. [11]	×	✓	✓	sym. transform loss [21]	Transformer	Umeyama [22]
2023	Remus et al. [23]	✓	✓	×	none	Encoder-decoder CNN	Direct regression
2024	Ikeda et al. [3]	✓	✓	✓	probabilistic	Diffusion model	TEASER++ [24]
2024	Fan et al. [15]	✓	✓	✓	none	Encoder-decoder CNN	Umeyama [22]
2024	Krishnan et al. [12]	✓	×	✓	none	Transformer	Direct regression

IV. 6D POSE SOLVER

None of the selected papers directly regress the 6D, 7D, or 9D pose from the input without intermediate steps. Regressing object poses without intermediate representations was shown to be inefficient [28]. All selected papers first predict an intermediate representation and solve the 6D pose subsequently. The papers [2], [7], [8], [9], [10], [11], [14], [3], [12] use NOCS as intermediate representation while [4] use an adapted version of NOCS, namely the Semantically-aware Object Coordinate Space (SOCS), a representation similar to NOCS with additional parameters to highlight semantically meaningful regions around keypoints. [13] perform pointcloud reconstruction and regress $\mathbf{R}$, $\mathbf{t}$, and s directly from latent features of the encoder-decoder network.

For networks that estimate the NOCS (or SOCS) pointcloud $\mathbf{P}_N$ in normalized space N, a final step for transforming $\mathbf{P}_N$ to the metric space M is needed to acquire $\mathbf{P}_M$. This step involves solving the equation: $\mathbf{P}_M = s \cdot \mathbf{R} \cdot \mathbf{P}_N + \mathbf{t}_M$, where $\mathbf{R}$, $\mathbf{t}_M$, and $\mathbf{s}$ are unknown. The authors of [2], [7], [8], [9], [11], [15] use the Umeyama algorithm to solve for $\mathbf{R}$, $\mathbf{t}_M$, and $\mathbf{s}$, and use metric depth data from a sensor to acquire $\mathbf{P}_M$. [8] uses a Metric Scale Object Shape (MSOS) branch to estimate a metric 3D model parallel to predicting $\mathbf{I}_N$. Therefore, they predict both $\mathbf{I}_N$ and $\mathbf{P}_M$ without using depth data from a sensor. The independence from depth comes with the drawback of increased runtime and limited performance since the MSOS branch can only interpolate between object models encountered during training. Objects with measurements beyond the ones seen in the training data (e.g., a realistic model car, vastly smaller than an actual car but with the same semantic properties) will lead to wrong results. [3] use the TEASER++ algorithm to solve for $\mathbf{R}$, $\mathbf{t}_M$, and s, and use depth data from a sensor to acquire $\mathbf{P}_M$. [12] directly regress $\mathbf{R}$ from $\mathbf{I}_N$, and regress $\mathbf{t}_M$ and $\mathbf{s}$ after the Dense Prediction Transformer (DPT) backbone [29]. [12] predict NOCS from a full scene with full semantic context, which helps to learn metric object size without depth data. Still, object sizes outside of the distribution seen during training will be challenging for such a model, similar to [8]. In regards to generalization, the deterministic algorithms Umeyama and TEASER++ do have the advantage of object category agnosticity. While [12] performs direct regression for 6D pose solving, added depth and a deterministic pose solver would most likely result in improved performance.

V. ROTATIONAL SYMMETRY HANDLING

While the pose of the textured object is distinct in all four views, the pose of the textureless object is not. The selected papers of this comparative study handle symmetry differently. Besides no explicit symmetry handling, three major symmetry handling techniques are used.

A. No explicit symmetry handling

[8], [9], [4], [14], [15], [12] do not mention any implicit nor explicit symmetry handling technique. Pose estimation performance is reported for the whole dataset, not single object categories.

B. Orthogonal vectors

[13] do not predict $\mathbf{R}$ as a 3x3 matrix but instead use two decoders that estimate two perpendicular vectors to describe $\mathbf{R}$. For objects with continuous symmetries around one axis the loss weight for one of the two perpendicular vectors is set to 0. This eliminates the constraint of predicting a fully defined pose, but is only applicable to continuous symmetries. Since this form of symmetry handling is explicit, symmetry types have to be manually annotated. For handling discrete symmetries [13] use the symmetry transform loss [21] described in the following section.

C. Symmetric transform loss

To handle discrete and continous symmetries [2], [7], [13], [11], [10] use the symmetric transform loss described by [21]. The symmetric transform loss $\mathscr{L}_{\text{sym}}$ can be defined as:

$$\mathscr{L}_{\text{sym}} = \min_{\mathbf{R} \in \mathscr{S}} \mathscr{L}\left(\mathbf{P}_{\text{est}}, \mathbf{R} \cdot \mathbf{P}_{\text{gt}}\right),$$

where:

- $\mathbf{P}_{\text{est}}$ is the estimated NOCS point cloud.

https://doi.org/10.34749/3061-0710.2025.13

Creative Commons Attribution 4.0 International License

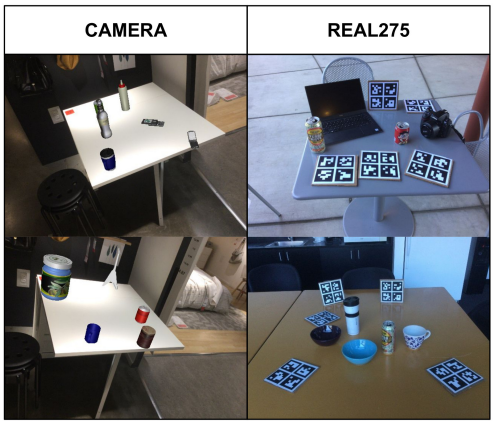

Fig. 3. **Exemplary images of the CAMERA and REAL275 datasets.** While the CAMERA dataset features rendered object instances on real backgrounds, the REAL275 dataset solely features real image data.

- $\mathbf{P}_{\text{gt}}$ is the ground truth NOCS point cloud.
- $\mathscr{S} = \{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_n\}$ is the set of all rotational symmetry transformations.
- $\mathscr{L}(\mathbf{P}_1, \mathbf{P}_2)$ is the loss function measuring the distance between two point clouds.

The symmetry transform loss requires handcrafted annotations of symmetrical poses for all objects in the training set. Annotation requires the symmetries to be discrete (e.g., a rotational symmetry has to be divided into n discrete symmetries around that axis). Furthermore, since real-life objects are either reconstructed or modeled with CAD, an arbitrary choice of symmetry is necessary. Reconstructed meshes are never fully symmetric due to reconstruction noise, and CAD parts may break symmetry only by a minor part of the object in relation to the full size. While these alterations technically break the symmetry, defining them as symmetric may still lead to better convergence when predicting NOCS.

D. Probabilistic symmetry handling

[3] presents an implicit symmetry handling approach for learning pose probabilities by sampling an additional noise input to the diffusion model. Compared to the other papers, they sample multiple noise inputs for each training sample drawn from the dataloader. Consequently, each instance of the training set has newly generated noise samples in each epoch during training. After predicting the NOCS the inlier rate of the TEASER++ pointcloud registration result is used as an additional loss for backpropagation. An example of this noise sampling approach is illustrated in Fig. 4. The advantage of this type of symmetry handling is that no discrete symmetry annotations are necessary.

VI. EXPERIMENTS

The selected papers report results on the REAL275 and Context-Aware MixEd ReAlity (CAMERA) datasets. Both datasets were introduced in [2]. Exemplary images are shown

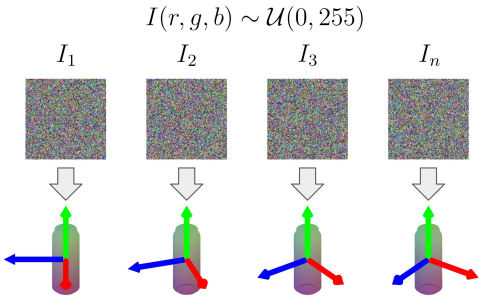

Fig. 4. **Illustration of the probabilistic loss used by DiffusionNOCS.** For each sample around the symmetry axis r,g,b values from a uniform Gaussian distribution are sampled. This introduces an adversarial character to the NOCS prediction, prohibiting the network from converging to local minima that do not capture the full symmetry of the object. By introducing this adverarial effect the network learns the symmetry of the object implicitly.

in Fig. 3. The CAMERA dataset consists of 300k synthetically rendered object images pasted onto real backgrounds. The REAL275 dataset contains 2750 test images across 18 different scenes. Both datasets contain the same object instances of 6 different object categories (bottle, bowl, camera, can, laptop, and mug), consisting of non-rotation-symmetric (camera, laptop, mug) and continuously symmetric (bottle, bowl, can) objects. Since the CAMERA dataset also uses synthetic object renderings for the test data, its relevance for evaluating algorithms for real-world pose estimation capabilities is questionable. If the test data is synthetic, the performance of algorithms on the real-world domain can not be evaluated.

Regarding accuracy metrics, the authors of [2], [7], [8], [12] report the mean Average Precision (mAP) of $3D_{25}$, $3D_{50}$, and $3D_{75}$ scores, which represent the Intersection over Union (IoU) between ground truth and estimated 3D bounding box at 25%, 50%, and 75% respectively. The authors of [2], [7], [8], [3] papers also report the mAP below error thresholds of **R** and **t**. The specific thresholds in the papers vary. Therefore, the most common ones are selected ($5°5cm$, $10°5cm$, and $10°10cm$).

VII. RESULTS

Table II shows the results presented in the selected papers on the CAMERA and REAL275 datasets. Out of all, the two papers that use solely RGB data, [8], [12] perform worst. This confirms the hypothesis that regressing object size and translation without depth data during runtime is challenging. While [8] report 75.4% of 3D bounding boxes within an IoU of 25%, the numbers drop to 32.4% at an IoU of 50% and 5.1% at an IoU of 75%. An opposite trend can be observed for the methods using solely depth. The best performing methods for 3D bounding box and **R**, **t** prediction rely on depth only. Especially [4], [14] appear to be the best performing methods on the REAL275 dataset for predicting **R**, **t**. [4] is slightly in the lead with 56.0% mAP $5°5cm$ and 82.0% $10°5cm$. [14] is close with 54.7% mAP $5°5cm$ and 81.6% $10°5cm$. Both [4], [14] do not perform explicit symmetry handling. While performance

https://doi.org/10.34749/3061-0710.2025.13

Creative Commons Attribution 4.0 International License

TABLE II
EVALUATION ON THE CAMERA AND REAL275 DATASETS. MAP SCORES FOR 3D BOUNDING BOXES, **R** AND **t**. "-" INDICATES SCORES NOT REPORTED IN THE RESPECTIVE PAPERS. BEST PERFORMANCE IN **BOLD**, WORST PERFORMANCE / RGB-ONLY UNDERLINED
† USE SYNTHETICALLY RENDERED TRAINING DATA, EVALUATE ON REAL TEST DATA.

Dataset	Year	Paper	Input	Priors	mAP					
					$3D_{25}$	$3D_{50}$	$3D_{75}$	$5°5cm$	$10°5cm$	$10°10cm$
CAMERA	2019	Wang et al. [2]	RGB-D	end-to-end	**91.1**	83.9	69.5	40.9	64.6	65.1
	2020	Tian et al. [7]	RGB-D	Mask R-CNN [30]	-	93.2	83.1	59.0	81.5	-
	2021	Lee et al. [8]	RGB	Mask R-CNN [30]	75.4	32.4	5.1	-	-	19.2
	2021	Chen et al. [13]	D	YOLOv3 [31]	-	-	85.2	62.0	-	-
	2022	Wang et al. [9]	RGB-D	Mask R-CNN [30]	-	**93.8**	88.0	76.4	87.7	-
	2023	Zhang et al. [10]	D	Mask R-CNN [30]	-	-	86.8	75.5	87.4	-
	2023	Wan et al. [4]	D	Mask R-CNN [30]	-	-	-	-	-	-
	2023	Wang et al. [14]	D	Mask R-CNN [30]	-	92.3	88.6	**83.9**	**90.5**	-
	2023	Zou et al. [11]	D	Mask R-CNN [30]	-	92.5	86.9	76.5	88.7	**89.9**
	2024	Ikeda et al. [3]	RGB-D	Mask R-CNN [30]	-	-	-	-	-	-
	2024	Fan et al. [15]	RGB-D	Mask R-CNN [30]	-	93.7	**89.6**	75.1	89.5	-
	2024	Krishnan et al. [12]	RGB	Mask R-CNN [30]	-	-	-	-	-	-
REAL275	2019	Wang et al. [2]	RGB-D	end-to-end	84.9	80.5	30.1	10.0	26.7	26.7
	2020	Tian et al. [7]	RGB-D	Mask R-CNN [30]	-	77.3	53.2	21.4	54.1	-
	2021	Lee et al. [8]	RGB	Mask R-CNN [30]	62.0	23.4	3.0	-	-	9.6
	2021	Chen et al. [13]	D	YOLOv3 [31]	**95.1**	**92.2**	63.5	28.2	60.8	64.6
	2021	Wang et al. [9]	RGB-D	Mask R-CNN [30]	-	79.3	55.9	34.3	60.8	-
	2022	Zhang et al. [10]	D	Mask R-CNN [30]	84.0	81.1	52.0	33.9	69.1	71.0
	2023	Wan et al. [4]	D	Mask R-CNN [30]	-	82.0	75.0	**56.0**	**82.0**	-
	2023	Wang et al. [14]	D	Mask R-CNN [30]	-	82.9	**76.0**	54.7	81.6	-
	2023	Zou et al. [11]	D	Mask R-CNN [30]	-	82.0	70.4	53.8	77.7	**79.8**
	2024	Ikeda et al. [3]†	RGB-D	Mask R-CNN [30]	-	-	-	35.0	66.6	-
	2024	Fan et al. [15]	RGB-D	Mask R-CNN [30]	-	82.3	66.6	41.3	67.0	-
	2024	Krishnan et al. [12]	RGB	Mask R-CNN [30]	43.5	10.6	-	-	-	-

metrics for individual object categories are missing in the papers, it appears that the 3D GCN of [4] and the CNNs + MLP network of [14] handle the symmetries of objects well enough. Both papers also use learning-based 6D pose solving, namely anisotropic scaling [4] and direct regression [14]. This sophisticated 6D pose solving techniques could be the reason for the superior performances. On the other hand, [11] comes close to the results of [4], [14] while employing the deterministic Umeyama method [22] for rigid point cloud alignment. The probabilistic loss of [3] does not lead to improved performance. However, since the authors of [3] use synthetic renderings for training and real data for evaluation, a fair comparison is not possible.

When comparing pose estimation performance, the source of object detection priors have to be taken into account. While [7], [8], [9], [10], [4], [14], [11], [3], [15], [12] use pre-computed Mask R-CNN location priors to ensure fair comparison, [2] use the location priors of their end-to-end trainable network and [13] use YOLOv3 for location priors. This results in superior results for 3D bounding box estimation on REAL275 by [13].

Overall, results are better on the CAMERA dataset compared to the REAL275 dataset. This can likely be attributed to the stronger domain shift between the training and test data of REAL275. Not only objects but also scenes are different, including other lighting, shadow and reflection. The domain shift for pose estimation only consists of object difference between training and test dataset.

Lastly, a clear correlation between improved performance and newer network architectures such as transformers or diffusion models cannot be observed.

VIII. CONCLUSION

This paper compares category-level object pose estimation methods which are evaluated on the CAMERA and REAL275 datasets. The methods differ regarding input modalities, symmetry handling, network types, and 6D pose solver algorithms. A comprehensive comparison was conducted, focusing on symmetry handling and its potential impact on model performance. After reviewing input modalities, network architectures, 6D pose solvers, symmetry handling, experiments, and results, the following conclusions are drawn:

- Omitting depth as done by [8], [12] drastically reduces performance as compared to RGB and RGB-D based methods.
- While the absence of depth data worsens results, the depth-only methods perform best overall, indicating that depth data is crucial for improving category-level object pose estimation.
- The two best performing methods use no explicit symmetry handling, suggesting that implicit symmetry handling is not mandatory if model architecture is allowing for it.
- While methods with learning-based 6D pose solvers excel regarding pose estimation performance, papers using deterministic methods such as Umeyama [22] achieve results almost on par. This is crucial since bridging category-level and open-set pose estimation benefits

https://doi.org/10.34749/3061-0710.2025.13

Creative Commons Attribution 4.0 International License

from deterministic 3D geometry-agnostic algorithms for 6D pose solving.
- The usage of different 2D object detection priors hinders a fair comparison, since improvement cannot clearly attributed to either detection or pose estimation.

Future research should build upon this comparative paper by analyzing the individual aspects of category-level object pose estimation further.

ACKNOWLEDGMENT

This research is supported by the EU program EC Horizon 2020 for Research and Innovation under grant agreement No. 101017089, project TraceBot, and the Austrian Science Fund (FWF), under project No. I 6114, iChores.

REFERENCES

[1] S. Thalhammer, D. Bauer, P. Hönig, J.-B. Weibel, J. García-Rodríguez, and M. Vincze, "Challenges for monocular 6-d object pose estimation in robotics," *IEEE Transactions on Robotics*, vol. 40, pp. 4065–4084, 2024.

[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] T. Ikeda, S. Zakharov, T. Ko, M. Z. Irshad, R. Lee, K. Liu, R. Ambrus, and K. Nishiwaki, "Diffusionnocs: Managing symmetry and uncertainty in sim2real multi-modal category-level pose estimation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 7406–7413.

[4] B. Wan, Y. Shi, and K. Xu, "Socs: Semantically-aware object coordinate space for category-level 6d object pose estimation under large shape variations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 14 065–14 074.

[5] P. Ausserlechner, D. Haberger, S. Thalhammer, J.-B. Weibel, and M. Vincze, "Zs6d: Zero-shot 6d object pose estimation using vision transformers," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 463–469.

[6] E. P. Örnek, Y. Labbé, B. Tekin, L. Ma, C. Keskin, C. Forster, and T. Hodan, "Foundpose: Unseen object pose estimation with foundation features," in *European Conference on Computer Vision*. Springer, 2024, pp. 163–182.

[7] 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.

[8] 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.

[9] J. Wang, K. Chen, and Q. Dou, "Category-level 6d object pose estimation via cascaded relation and recurrent reconstruction networks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4807–4814.

[10] R. Zhang, Y. Di, F. Manhardt, F. Tombari, and X. Ji, "Ssp-pose: Symmetry-aware shape prior deformation for direct category-level object pose estimation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7452–7459.

[11] L. Zou, Z. Huang, N. Gu, and G. Wang, "Gpt-cope: A graph-guided point transformer for category-level object pose estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[12] A. Krishnan, A. Kundu, K.-K. Maninis, J. Hays, and M. Brown, "Omninocs: A unified nocs dataset and model for 3d lifting of 2d objects," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2024, pp. 127–145.

[13] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1581–1590.

[14] R. Wang, X. Wang, T. Li, R. Yang, M. Wan, and W. Liu, "Query6dof: Learning sparse queries as implicit shape prior for category-level 6dof pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 14 055–14 064.

[15] Z. Fan, Z. Song, Z. Wang, J. Xu, K. Wu, H. Liu, and J. He, "Acr-pose: Adversarial canonical representation reconstruction network for category level 6d object pose estimation," in *Proceedings of the International Conference on Multimedia Retrieval (ICMR)*. Association for Computing Machinery, 2024, p. 55–63.

[16] K. Chen, S. James, C. Sui, Y.-H. Liu, P. Abbeel, and Q. Dou, "Stereopose: Category-level 6d transparent object pose estimation from stereo images via back-view nocs," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2855–2861.

[17] M. Zaccaria, F. Manhardt, Y. Di, F. Tombari, J. Aleotti, and M. Giorgini, "Self-supervised category-level 6d object pose estimation with optical flow consistency," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2510–2517, 2023.

[18] G. Marullo, L. Tanzi, P. Piazzolla, and E. Vezzetti, "6d object position estimation from 2d images: A literature review," *Multimedia Tools and Applications*, vol. 82, no. 16, pp. 24 605–24 643, 2023.

[19] Z. Fan, Y. Zhu, Y. He, Q. Sun, H. Liu, and J. He, "Deep learning on monocular object pose detection and tracking: A comprehensive overview," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–40, 2022.

[20] D. Bauer, T. Patten, and M. Vincze, "Verefine: Integrating object pose verification with physics-guided iterative refinement," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4289–4296, 2020.

[21] G. Pitteri, M. Ramamonjisoa, S. Ilic, and V. Lepetit, "On object symmetries and 6d pose estimation from images," in *Proceedings of the International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 614–622.

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

[23] A. Remus, S. D'Avella, F. D. Felice, P. Tripicchio, and C. A. Avizzano, "i2c-net: Using instance-level neural networks for monocular category-level 6d pose estimation," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1515–1522, 2023.

[24] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2020.

[25] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[27] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[28] 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.

[29] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF Iinternational Conference on Computer Vision (ICCV)*, 2021, pp. 12 179–12 188.

[30] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow," *GitHub repository*, 2017.

[31] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018. [Online]. Available: https://arxiv.org/abs/1804.02767

https://doi.org/10.34749/3061-0710.2025.13

Creative Commons Attribution 4.0 International License