GIVEPose: Gradual Intra-class Variation Elimination for RGB-based Category-Level Object Pose Estimation —Supplementary Material

Ziqin Huang[1] Gu Wang[1] Chenyangguang Zhang[1] Ruida Zhang[1]
Xiu Li[1,2] Xiangyang Ji[1]
[1]Tsinghua University [2]Pengcheng Laboratory
{huang-zq24@mails., xyji@}tsinghua.edu.cn

Contents

A. More Details of Loss Functions

As shown in Sec. 3.5 of the main paper, our overall loss function consists of three components: L_{pose}, L_{nocs}, and L_{ivfc}. Here, we present the detailed formulation of each loss term.

A.1. Pose losses

Following LaPose [10], we employ similar loss functions to supervise pose regression. The overall pose loss is

$$\begin{aligned} L_{pose} = \omega_{rot} L_{rot} + \omega_{pm} L_{pm} \\ + \omega_{trans} L_{trans} + \omega_{size} L_{size}, \end{aligned} \tag{A.1}$$

where ω_{rot}, ω_{pm}, ω_{trans}, and ω_{size} are the weighting hyper-parameters.

Specifically, L_{trans} and L_{size} are utilized to supervise the scale-agnostic translation and size, respectively:

$$\begin{aligned} L_{trans} &= \|\hat{t} - t_{gt}\|_1, \\ L_{size} &= \|\hat{s} - s_{gt}\|_1, \end{aligned} \tag{A.2}$$

where $\hat{t}$ and $\hat{s}$ represent the predicted scale-agnostic translation and size respectively, t_{gt} and s_{gt} denote corresponding ground truth value.

Both L_{rot} and L_{pm} serve as supervision terms for rotation learning, where L_{rot} directly supervises the rotation matrix, and the point matching loss L_{pm} [6] is calculated by first applying rotational transformations to the points on the model:

$$\begin{aligned} L_{rot} &= \|\hat{R} - R_{gt}\|_1, \\ L_{pm} &= \text{avg}_{x \in \mathbf{M}} \|\hat{R}x - R_{gt}x\|_1, \end{aligned} \tag{A.3}$$

where R_{gt} represents the ground-truth rotation matrix for supervising the predicted $\hat{R}$, and x denotes the sampled points from the object's NOCS model $\mathbf{M}$. To handle ambiguous rotations arising from object symmetry [5], we supervise the prediction using the closest rotation selected from the proper symmetry group for symmetrical categories.

A.2. Geometric losses in GIVE

For the implementation of our proposed GIVE strategy, we utilize L_{nocs} and L_{ivfc} to supervise two intermediate representations, the NOCS map and the IVFC map:

$$L_{nocs} = \|M_{nocs} \cdot (N_{map}^{gt} - \hat{N}_{map})\|_1, \tag{A.4}$$

where M_{nocs} represents the mask of the NOCS map and the N_{map}^{gt} is the supervision of predicted NOCS map $\hat{N}_{map}$.

$$L_{ivfc} = \|M_{ivfc} \cdot (C_{map}^{gt} - \hat{C}_{map})\|_1, \tag{A.5}$$

where M_{ivfc} denotes the mask of the IVFC map and the prediction of IVFC map $\hat{C}_{map}$ is supervised by corresponding ground-truth value C_{map}^{gt}.

B. Details of Deformable Convolutional Auto-Encoder (DCAE)-based module

Throughout the network architecture, the DCAE-based module serves as the key component for achieving gradual intra-class variation elimination.

In our implementation, we adopt the deformable convolution (specifically, DCNv3) proposed in [8]. As illustrated in Fig. A.1, we employ three layers of deformable convolutions with a stride of 2 to extract features and reduce the resolution from the predicted NOCS map, resulting in a feature map of size $256 \times 8 \times 8$. Compared to vanilla convolutions, deformable convolutions feature adaptable convolution kernels, enabling more flexible spatial correspondence between feature maps and input coordinate maps. This flexibility allows for a more robust capture of category-consensus information from the NOCS map. The extracted features are subsequently concatenated with backbone features. Finally, the concatenated feature map undergoes three upsampling operations, consisting of one deconvolution and two bilinear interpolations, to generate the IVFC map with redundant instance information eliminated.

C. Details of Category-Consensus Model Reconstruction

As mentioned in Sec. 4.2 of the main paper, to obtain the mesh models for generating IVFC maps, we performed surface reconstruction on per-category mean point cloud models using functions from the Open3D [11] library. Specifically, we first reconstructed mesh models directly from point clouds using the $create\_from\_point\_cloud\_alpha\_shape$ function, followed by applying Laplacian smoothing through the $filter\_smooth\_laplacian$ function. Following the approach in NOCS [10], we further color-coded the mesh models based on their coordinate values to obtain the color-coded category-consensus model, as shown in Fig. C.2.

D. Extended Quantitative Evaluation

D.1. Per-category results

Tab. D.1, Tab. D.2, and Tab. D.3 present the detailed evaluation results of our GIVEPose for each category on REAL275, CAMERA25 [7], and Wild6D [1] datasets using scale-agnostic evaluation metrics [10].

D.2. Detailed Performance Comparison of Our Method against LaPose

For a more detailed comparison with existing methods, we provide per-category results of our method and LaPose [10] in Fig. D.3. As illustrated in the figure, our method shows significant improvements over LaPose [10] for categories with large intra-class variation (e.g., camera), which can be attributed to our gradual intra-class variation elimination strategy. For the bottle category, our method experiences a slight performance drop, which we attribute to the joint training of multiple categories, where inter-category influences may occur. Overall, our method outperforms LaPose [10], achieving better average results and superior performance across most categories, thereby demonstrating its effectiveness.

E. Extended Qualitative Analysis

To facilitate a more comprehensive and intuitive evaluation, we present extended qualitative comparisons between our approach and four existing methods [2, 3, 9, 10] in Fig. G.4 and Fig. G.5. These results demonstrate that our method achieves more accurate category-level pose estimation compared to existing approaches, particularly when handling categories with large intra-class variations and processing truncated cropped images.

F. Code and Reproducibility

We provide our PyTorch-based [4] code in the "GIVEPOSE" folder for anonymous review, with detailed instructions for reproducing experimental results available in the "GIVEPOSE/README.md" file. The code will be publicly released upon acceptance.

G. Ethics Statement

Despite the consideration of scene and object diversity in the publicly available datasets we used, potential biases continue to exist. We acknowledge the significant computational resources and energy consumption required for training and inference, which could raise environmental concerns. Efforts are being made to minimize these impacts through efficient computation strategies and exploring more sustainable AI practices. Furthermore, all datasets were used in compliance with ethical standards, ensuring data privacy.

References

[1] Yang Fu and Xiaolong Wang. Category-level 6d object pose estimation in the wild: A semi-supervised learning approach and a new dataset. *NeurIPS*, 35: 27469–27483, 2022. 2, 4

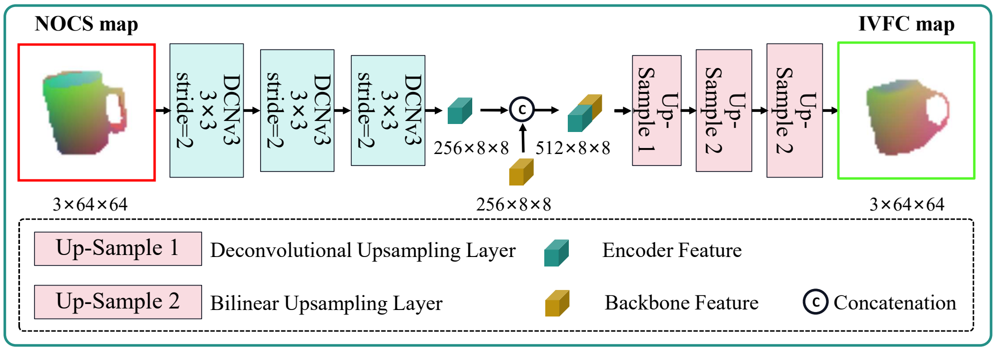

Figure A.1. Detailed structural diagram of the Deformable Convolutional Auto-Encoder (DCAE)-based module.

[2] Taeyeop Lee, Byeong-Uk Lee, Myungchul Kim, and In So Kweon. Category-level metric scale object shape and pose estimation. *IEEE RA-L*, 6(4):8575–8582, 2021. 2

[3] Jiehong Lin, Zewei Wei, Changxing Ding, and Kui Jia. Category-level 6d object pose and size estimation using self-supervised deep prior deformation networks. In *ECCV*, pages 19–34. Springer, 2022. 2

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 2

[5] Giorgia Pitteri, Michaël Ramamonjisoa, Slobodan Ilic, and Vincent Lepetit. On object symmetries and 6d pose estimation from images. In *3DV*, pages 614–622. IEEE, 2019. 1

[6] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *CVPR*, pages 16611–16621, 2021. 1

[7] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, pages 2642–2651, 2019. 2, 4

[8] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *CVPR*, pages 14408–14419, 2023. 2

[9] Jiaxin Wei, Xibin Song, Weizhe Liu, Laurent Kneip, Hongdong Li, and Pan Ji. Rgb-based category-level object pose estimation via decoupled metric scale recovery. In *ICRA*, 2024. 2

[10] Ruida Zhang, Ziqin Huang, Gu Wang, Chenyang-guang Zhang, Yan Di, Xingxing Zuo, Jiwen Tang, and Xiangyang Ji. Lapose: Laplacian mixture shape modeling for rgb-based category-level object pose estimation. In *ECCV*, 2024. 1, 2, 3

[11] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018. 2

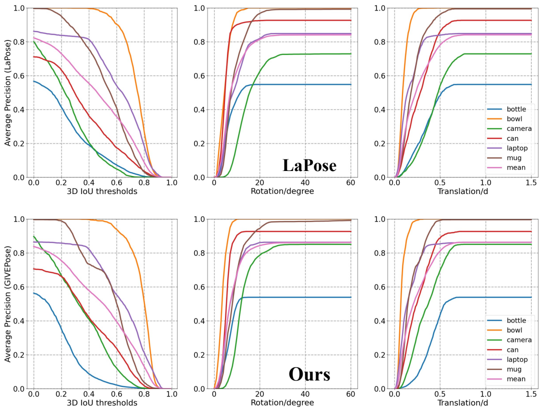

Figure D.3. Detailed Comparison with Lapose [10] on the REAL275 dataset using scale-agnostic evaluation metrics.

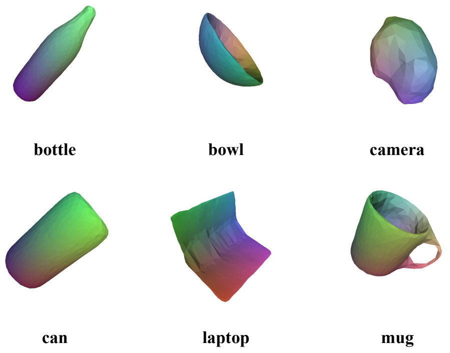

Figure C.2. Color-coded consensus models for each category.

Category	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$10°$
bottle	26.6	4.6	0.5	4.6	39.6	4.8	43.5	49.7
bowl	100.0	98.9	73.5	97.9	99.9	98.1	100.0	99.9
camera	60.2	26.3	1.3	10.2	27.8	21.1	69.8	30.8
can	61.3	32.8	6.1	41.7	86.7	41.9	88.7	90.4
laptop	85.4	70.2	33.2	56.6	66.6	61.8	85.4	67.2
mug	95.1	69.3	10.2	56.7	68.5	65.8	99.2	68.5
average	71.4	50.3	20.8	44.6	64.8	48.9	81.1	67.8

Table D.1. Per-category results on the REAL275 dataset using scale-agnostic evaluation metrics.

Category	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$10°$
bottle	78.0	57.6	20.4	56.1	82.6	56.2	83.2	87.4
bowl	95.4	86.7	35.6	82.5	95.6	82.9	96.3	95.9
camera	60.0	23.9	3.3	18.1	59.9	21.5	72.6	72.2
can	76.5	52.1	13.4	43.6	78.5	43.6	78.5	87.2
laptop	92.1	74.8	39.3	68.2	89.5	72.0	95.6	91.7
mug	54.8	24.8	4.2	16.3	7.2	22.1	64.1	59.8
average	76.1	53.3	19.4	47.5	75.5	82.4	49.7	81.7

Table D.2. Per-category results on the CAMERA25 dataset [7] using scale-agnostic evaluation metrics.

Category	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$10°$
bottle	87.3	62.4	17.4	57.1	76.7	68.9	94.9	77.3
bowl	99.3	91.8	42.9	87	96.2	88.5	99.7	96.3
camera	60.8	22.0	0.3	1.2	2.8	26.4	73.4	3.0
laptop	99.7	99.6	79.9	18	18.1	99.4	99.7	18.1
mug	89.6	26.4	0.1	3.9	7.5	24	93.7	7.5
average	87.3	60.4	28.1	33.4	40.3	61.4	92.3	40.4

Table D.3. Per-category results on the Wild6D dataset [1] using scale-agnostic evaluation metrics.

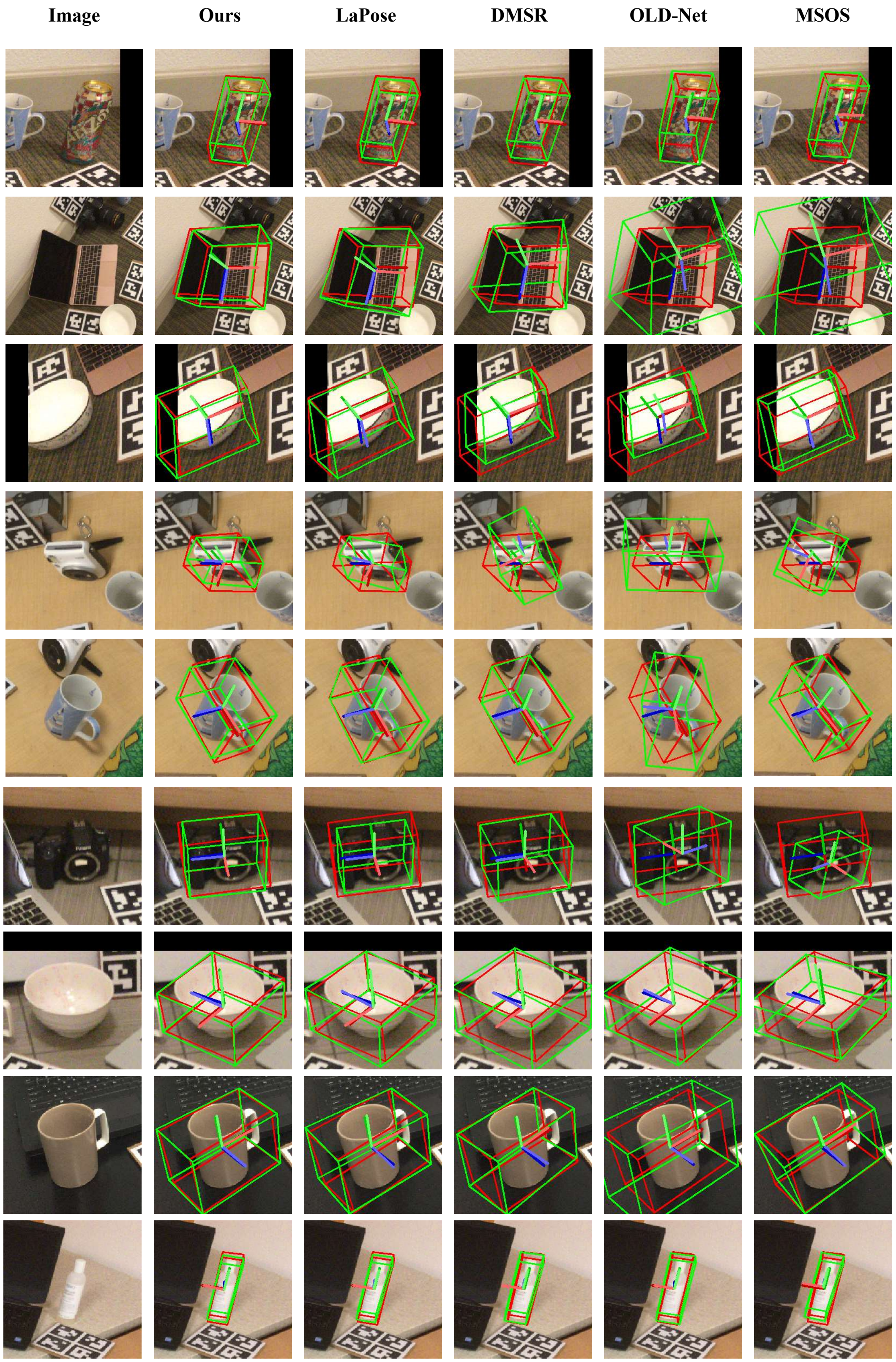

Figure G.4. Qualitative comparisons on **REAL275**. For the 3D box visualization, red denotes the ground truth and green represents the predicted result. For the axis projections, darker shades indicate the ground truth, while lighter shades correspond to the predicted results.

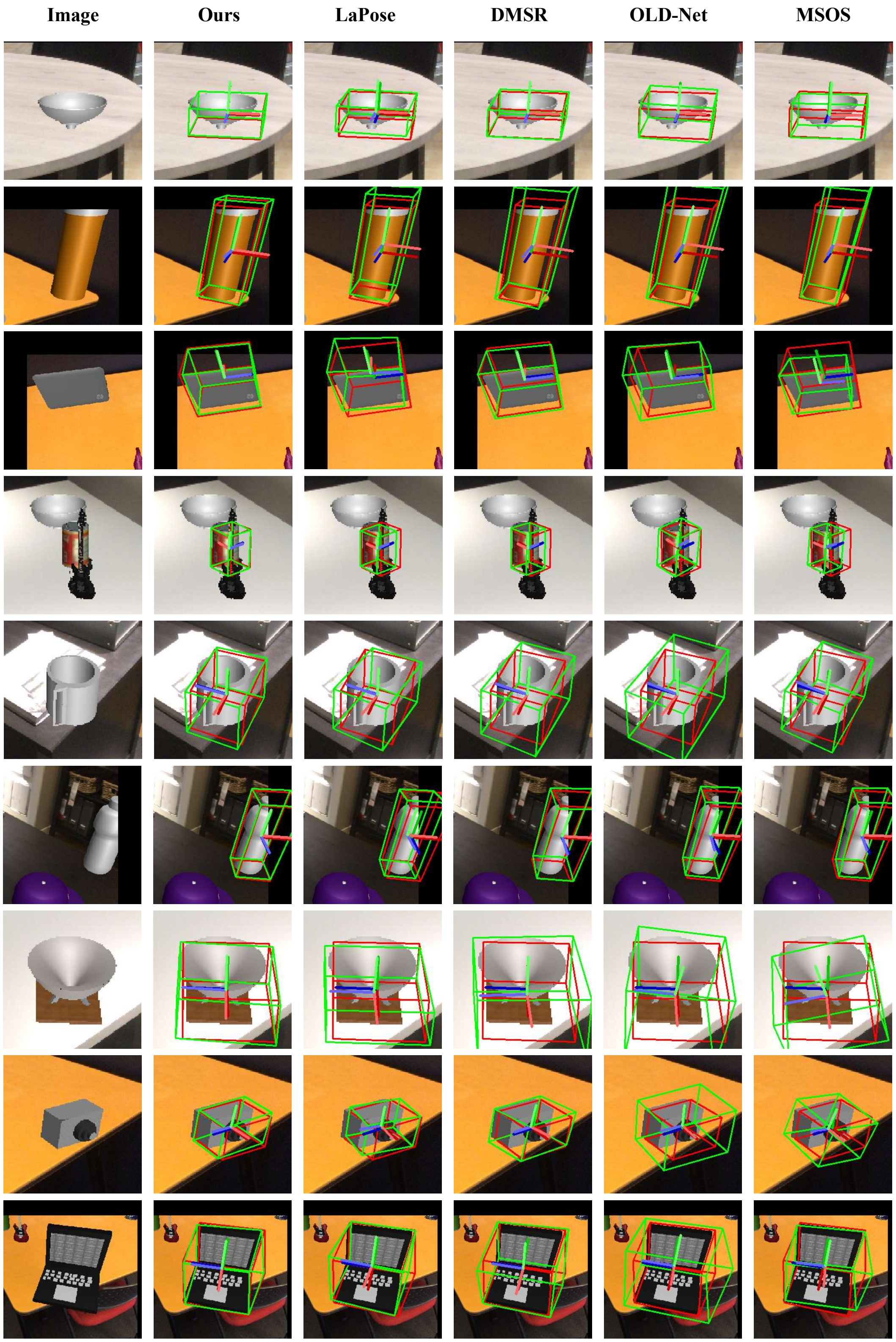

Figure G.5. Qualitative comparisons on **CAMERA25**. For the 3D box visualization, red denotes the ground truth and green represents the predicted result. For the axis projections, darker shades indicate the ground truth, while lighter shades correspond to the predicted results.