Unified Category-Level Object Detection and Pose Estimation from RGB Images using 3D Prototypes

Supplementary Material

A. Further Implementation Details

In this section, we provide additional details on the training regimen of our method.

Parameters. We train the model on a single NVIDIA H100 GPU for 150k iterations using the *AdamW* [6] optimizer with weight decay of 0.05 and a Cosine Annealing schedule [5] that decays the learning rate from $1e-4$ to $1e-7$. As training and inference-specific hyperparameters, we choose $\kappa = \frac{1}{0.07}$, $t_1 = 0.5$, and $t_2 = 0.7$. The adapters are included in each transformer block of the DINOv2 [7] model with a low-rank dimensionality of 128.

Training Data. To improve generalization, we use a similar data augmentation strategy during training as [1]. In each training iteration, we use a batch size of 10 images which are stacked into dynamic batches according to the present objects.

Annotation Generation. We use PyTorch3D [8] to rasterize the object prototypes into the annotation set. For each object, we render the given pose individually and then extract the per-pixel annotations. To account for objects occluding each other (which happens frequently for the category-level feature map) we mask out pixels where more than one object is visible by setting the visibility to 0. Alternatively, one could also opt for supervising with the closest object. However, we found that this results in the model to miss small, or partially visible objects more frequently.

B. Architectural choices

We show an ablation on the core design choices of our method in Tab. 1.

Adapter. The PEFT strategy to introduce dataset- and task-specific information into the pre-trained feature extractor massively benefits our method. Only by modifying the feedforward part of the transformer blocks shows significant performance improvements which is especially noticeable for rotation accuracy.

Foreground Modeling. Next, we evaluated the effect of our foreground modeling via CrossAttention. Specifically, we compare two variants. First, we evaluate the effect of focusing the model onto the foreground region during training via a baseline that has all CrossAttention layers removed and filters outliers during inference with confidence scores. We found that the same threshold $t_2 = 0.7$ we used for the full model is not ideal in this case and a more robust segmentation is obtained with $t_2 = 0.8$. This naive baseline achieves good results, indicating that our method can identify vertices with high likelihood. Next, we use our full model but ignore the provided mask during inference by setting $t_1 = 0$. This variant consistently outperforms the previous, indicating that focusing the model on the foreground region explicitly during training leads to better representation learning. However, utilizing the foreground mask still provides consistent improvements across all metrics, indicating its importance for correspondence estimation.

Pose refinement. Finally, we ablate over the components in or 9D pose refinement stage that utilizes the features that follow the instance-level prototype geometries. We show that returning the poses obtained from ProgX (i.e. 6D poses) leads to consistently worse performance. Even an instance level refinement using the category-level correspondences $\bar{\mathcal{N}}_{3D}^{2D}$ leads to performance improvements (see row "w refinement"). However, to obtain strong results for the tightest bounding box threshold $NIoU_{75}$ the size optimization using the instance-level correspondences $\mathcal{N}_{3D}^{2D}$ is required (see row "w size estimation"). Best performance, however, is obtained when refining the deformed 2D/3D correspondences, leading to our full pipeline.

C. Inference Speed

We compare the inference speed of our method with the two-stage baselines in Fig. 1. In contrast to baselines, our method requires only a single forward pass. Two-stage methods require one forward pass of the detection model and one call for each detected object. Our method, on the other hand, is more reliant on the choice of output resolution and found correspondences. With a more aggressive outlier rejection or subsampling of correspondences inference speed can be greatly improved with only minor loss in accuracy. In this work, however, we did not optimize for inference speed and consider speed-up strategies as future work.

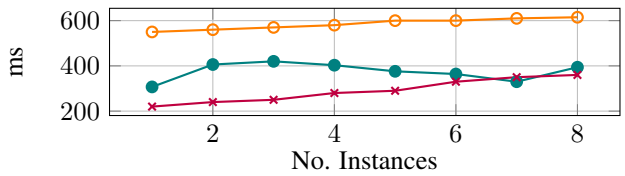

Figure 1. Average inference runtime our method and the baselines LaPose and DMSR. Runtime of our method depends on the found correspondences instead of present instances.

Method	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$5°0.2d$	$5°0.5d$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$5°$	$10°$
Ours	75.2	53.7	19.2	25.1	31.8	43.7	66.1	53.5	83.7	32.1	68.8
w/o adapter	-16.4	- 21.0	-15.9	-19.7	-22.1	-25.8	-31.1	-21.6	-8.7	- 21.8	-30.8
w/o CA - $t_1 = 0, t_2 = 0.7$	- 3.8	- 9.4	- 4.8	- 7.7	- 4.0	- 7.3	- 4.5	- 8.7	- 0.1	- 3.4	- 3.8
w/o CA - $t_1 = 0, t_2 =$opt.	+ 0.8	- 4.9	- 2.6	- 5.7	- 1.9	- 5.6	- 2.9	- 6.3	- 0.3	- 1.7	- 2.9
w CA - $t_1 = 0, t_2 = 0.7$	- 1.4	- 4.9	- 3.0	- 4.2	- 1.9	- 4.5	- 2.6	- 5.4	+ 0.2	- 1.8	- 2.1
w CA - $t_1 = 0, t_2 =$opt.	+ 0.6	- 0.9	- 0.7	- 0.9	- 0.3	- 1.9	- 1.2	- 1.7	- 0.2	- 0.3	- 1.2
w/o refinement + w/o size	- 0.7	- 2.7	- 3.9	- 2.1	- 2.1	- 2.4	- 3.9	- 2.1	- 1.0	- 1.8	- 3.5
w refinement	- 0.3	- 2.4	- 2.8	- 1.3	- 1.1	- 1.9	- 2.7	- 1.7	- 0.8	- 0.8	- 2.2
w size estimation	- 1.0	- 2.2	+ 0.1	- 2.4	- 2.4	- 2.5	- 3.9	- 2.2	- 0.9	- 2.2	- 3.6

Table 1. Ablation over the key design choices in Φ. The adapters are crucial for precise pose estimation and their removal leads to massive performance drops. In the second block, we evaluate without using the foreground mask obtained from the CrossAttention layers and solely from confidence values. "w/o CA" was trained without any CrossAttention layers, with the rest of the architecture being identical. Confidence measures alone lead to reasonable performance, especially with the optimal threshold parameter ($t_2 = 0.8$). However, it is still consistently worse than the network trained with CrossAttention ("w CA"), even without using the mask during inference when setting $t_1 = 0$. In the third block, we ablate over the choices in the instance-level 9D pose refinement part of our pipeline. "w/o refinement + w/o size" refers to directly outputting the poses after ProgX, yielding consistently worse results. "w refinement" refers to the instance level pose refinement given the category-level correspondences, which gives a marginal improvement across all metrics. "w size estimation" includes the size optimization from the instance-level correspondences and shows that this is crucial for good performance on the tight $NIoU_{75}$ metric.

D. Pose Estimation for Overlapping Objects

REAL275 [9] does not contain many overlapping objects of the same category. To showcase that our method can deal with intra-category overlaps we captured in-the-wild images with a smartphone and approximated its intrinsic matrix. We show the predictions of our method in Fig. 2.

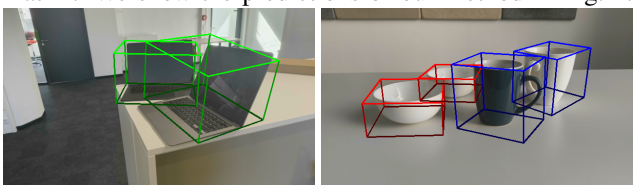

Figure 2. Detections and poses from our model with same-category occlusions on self-captured images.

E. Additional Quantitative Results on Robustness Study

In Tab. 2, we show the pose estimation accuracy under scale-agnostic metrics averaged over all corruption types. The corrupted images are generated using the method proposed in [3] and using their public code. We consider four types of image degradations, encompassing a total of eight corruption types: noise (speckle noise, Gaussian noise), blur (Gaussian blur, defocus blur), digital artifacts (JPEG compression, elastic transformation), and weather effects (frost, fog). The corruption strength follows the default setting, varying in severity per image basis. For a fair comparison, we run each method on the same set of images.

We report the mean scale-agnostic 3D Intersection over Union (NIoU), rotation, and translation metrics for all methods. Additionally, we show results for each corruption type at ROI level in Tab. 4 and at image level in Tab. 3.

F. Qualitative Results on Corrupted Images

Fig. 3 presents qualitative examples of object detection and pose estimation on the corrupted REAL275 dataset. We can observe that image degradations negatively affect the detection and in turn, the final pose estimation accuracy. For instance, when applying elastic transformations the reduced detector accuracy causes the laptop to be missed entirely and introduces redundant detections of the camera. This shows that for degraded images the detection model is a performance bottleneck in current two-stage approaches. In the single-stage approach, we benefit from significantly more robust detection and pose estimation quality.

G. Qualitative Results of Dense Matching

In Fig. 4, we present qualitative results of dense 2D-3D correspondence matching. We render all NOCS-maps with the geometry of our object prototypes using PyTorch3D [8]. To be consistent with our main inference, we remove correspondences with a confidence score below $t_2 = 0.7$. To address object overlap, we prioritize rendering the object closer to the camera. We observe that our method generates less confident correspondences near object edges, likely due to the ambiguity of these regions during optimization. Overall, our method produces high-quality correspondences using a simple nearest-neighbor matching approach, indicating that the contrastive training approach produces descriptive features.

Source	Method	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$5°0.2d$	$5°0.5d$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$5°$	$10°$
None	MSOS [4]	36.9	9.7	0.7	-	-	3.3	15.3	10.6	50.8	-	17.0
	OLD-Net [2]	35.4	11.4	0.4	0.9	3.0	5.0	16.0	12.4	46.2	4.2	20.9
	DMSR [10]	57.2	38.4	9.5	15.1	23.7	25.6	45.2	35.0	67.3	27.4	52.0
	LaPose [11]	70.7	47.9	15.8	15.7	21.3	37.4	57.4	46.9	78.8	23.4	60.7
	Ours	**71.6**	**50.5**	**18.3**	**21.6**	**28.6**	**41.9**	**61.6**	**51.0**	**80.9**	**29.1**	**64.1**
ROI	OLD-Net[2]	30.3	8.6	0.2	0.5	1.9	3.6	12.2	10.4	41.2	3.2	17.1
	DMSR[10]	55.3	35.6	8.2	12.4	19.4	23.5	40.5	33.4	65.5	22.6	46.0
	LaPose[11]	64.1	39.8	12.9	11.2	16.2	27.9	47.9	38.3	73.9	18.4	52.0
Image	OLD-Net[2]	25.5	7.0	0.2	0.4	1.7	2.7	9.4	8.1	34.0	3.2	15.7
	DMSR[10]	49.3	32.9	7.4	11.4	17.1	21.4	35.2	30.4	57.1	20.1	40.3
	LaPose[11]	54.5	36.1	12.2	11.1	15.2	26.4	41.5	34.9	62.1	17.3	45.0
	Ours	**63.8**	**43.4**	**14.3**	**17.9**	**24.7**	**35.2**	**53.7**	**44.5**	**73.2**	**25.2**	**56.0**

Table 2. Ablation study of robustness under image noises. We report the performance of all methods on clean data (top), as well as ROI corruptions for baselines (middle), and image-level corruptions (bottom). Note, that performance of our method on clean data is different due to the changed training regiment to make this comparison fair.

Method	Noise	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$5°0.2d$	$5°0.5d$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$5°$	$10°$
LaPose	Speckle Noise	56.7	35.8	**13.3**	8.6	11.7	25.1	39.8	34.9	64.9	13.9	44.2
DMSR		48.8	31.1	6.4	8.9	15.2	17.9	33.4	27.9	55.7	19.4	40.3
Old-Net		23.2	6.7	0.2	0.2	1.3	2.6	0.3	7.7	32.1	3.2	17.0
Ours		**61.8**	**40.3**	12.6	**15.0**	**22.6**	**31.1**	**50.6**	**40.8**	**71.6**	**23.2**	**53.5**
LaPose	Gaussian Blur	58.2	41.9	17.3	13.4	18.0	31.1	44.1	40.1	64.0	20.2	47.6
DMSR		54.3	40.4	10.3	14.1	18.5	27.7	39.4	36.6	60.3	21.2	43.5
Old-Net		30.7	7.4	0.3	0.4	2.6	3.3	13.2	9.7	39.1	4.1	18.5
Ours		**69.9**	**48.9**	**17.4**	**19.6**	**25.9**	**39.7**	**59.3**	**49.5**	**78.9**	**26.2**	**61.3**
LaPose	Gaussian Noise	52.2	32.5	11.9	8.3	11.3	23.9	36.4	31.9	60.8	13.0	40.0
DMSR		45.2	30.1	7.2	10.6	17.1	19.3	34.0	28.0	52.9	19.3	39.2
Old-Net		24.8	8.7	0.2	0.6	2.2	3.3	11.4	9.0	32.3	4.1	17.2
Ours		**56.9**	**37.8**	**12.2**	**14.9**	**21.0**	**30.4**	**47.1**	**38.7**	**65.4**	**21.7**	**49.7**
LaPose	Defocus Blur	54.1	39.3	11.4	12.2	16.1	29.0	42.1	36.9	60.2	17.4	44.3
DMSR		53.2	37.9	8.8	10.8	15.4	24.5	37.2	36	61.8	17.4	41.1
Old-Net		22.6	5.9	0.1	0.2	1.3	2.1	8.8	7.6	30.0	3.1	15.1
Ours		**63.2**	**42.3**	**14.7**	**17.8**	**24.6**	**34.6**	**53.5**	**43.1**	**73.0**	**24.9**	**55.9**
LaPose	JPEG Compression	57.3	37.6	9.9	11.1	16.6	27.1	45.1	36.1	66.9	19.3	49.2
DMSR		51.3	32.4	8.1	11.3	14.8	21.5	32.3	31.7	60.3	16.5	34.9
Old-Net		35.9	12.4	0.3	0.8	2.7	5.0	15.8	13.5	44.8	4.1	20.1
Ours		**70.4**	**50.7**	**18.5**	**22.1**	**29.2**	**41.4**	**59.5**	**51.6**	**80.2**	**29.7**	**62.4**
LaPose	Elastic Transform	58.6	36.9	10.5	12.0	17.3	27.7	47.1	35.2	66.4	19.5	50.8
DMSR		49.0	30.9	6.4	11.2	17.4	19.5	35.0	27.9	57.5	20.5	40.8
Old-Net		25.3	5.6	0.2	0.4	2.6	2.2	11.9	6.2	34.8	4.3	17.6
Ours		**68.0**	**45.4**	**15.3**	**17.8**	**24.7**	**36.9**	**57.9**	**45.9**	**78.5**	**25.1**	**60.5**
LaPose	Frost	49.4	32.3	11.5	11.6	15.1	23.7	38.5	31.9	56.8	17.7	42.0
DMSR		39.5	25.1	4.8	10.1	16.9	16.4	29.0	23.4	46.6	20.0	34.7
Old-Net		23.8	6.6	0.2	0.2	0.9	2.2	8.9	7.4	31.6	1.6	12.8
Ours		**60.9**	**41.1**	**12.8**	**18.4**	**25.4**	**33.9**	**51.2**	**43.8**	**70.7**	**25.7**	**53.0**
LaPose	Fog	**66.9**	**46.5**	**17.2**	15.2	19.9	**35.0**	**53.9**	**46.0**	**75.3**	22.0	**57.0**
DMSR		53.1	35.2	7.5	14.0	21.7	24.1	41.1	32.1	61.7	26.3	47.9
Old-Net		17.8	2.4	0.1	0.1	0.3	1.1	4.9	4.0	27.6	0.7	7.3
Ours		58.9	40.6	11.1	**17.4**	**24.4**	33.4	50.2	42.6	67.5	**24.9**	51.8

Table 3. We show per corruption accuracy of all methods. Corruptions are applied to the full image, affecting both detection and pose estimation. Our method outperforms the baselines on a majority of the corruption types.

Method	Noise	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$5°0.2d$	$5°0.5d$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$5°$	$10°$
LaPose	Speckle Noise	**62.2**	**36.5**	**12.6**	7.0	10.1	**23.7**	40.2	**34.5**	**71.6**	11.9	45.3
DMSR		54.9	35.4	8.9	**10.9**	**18.0**	22.9	**40.5**	32.8	65.4	**21.1**	**46.0**
Old-Net		30.9	9.1	0.2	0.5	2.0	4.1	13.4	10.6	41.6	3.6	19.1
LaPose	Gaussian Noise	**62.6**	**34.3**	**12.0**	7.6	10.8	**22.5**	37.2	**33.8**	**72.4**	12.3	41.5
DMSR		55.3	**34.3**	7.6	**11.8**	**19.0**	22.3	**39.5**	32.0	65.3	**21.4**	**44.9**
Old-Net		33.3	10.4	0.2	0.7	2.2	4.6	13.6	11.6	43.5	3.5	19.0
LaPose	Gaussian Blur	**61.8**	**41.5**	**15.3**	13.3	19.0	**31.4**	**49.9**	**40.5**	**71.9**	22.1	**54.5**
DMSR		57.9	39.8	10.1	**14.4**	**22.2**	26.7	45.1	37.5	67.7	**26.1**	51.4
Old-Net		30.6	7.6	0.3	0.5	2.9	3.5	13.7	9.9	41.5	4.6	18.9
LaPose	Defocus Blur	**60.8**	**38.8**	**11.0**	**11.3**	**16.7**	**27.3**	**48.0**	**36.1**	**71.3**	**18.8**	**51.8**
DMSR		56.3	37.9	8.7	10.6	15.7	24.0	38.9	35.8	66.2	18.5	44.0
Old-Net		25.9	6.2	0.1	0.2	1.3	1.9	8.6	9.0	36.6	3.4	15.7
LaPose	JPEG Compression	**58.3**	**34.2**	**8.9**	9.9	**16.3**	**24.6**	**46.3**	**33.3**	**70.5**	**19.1**	**51.3**
DMSR		50.6	29.9	6.4	**10.6**	14.9	20.4	33.5	29.1	61.9	16.7	36.9
Old-Net		35.7	11.5	0.4	0.9	2.9	5.0	15.8	12.9	46.5	4.3	20.4
LaPose	Elastic Transform	**69.1**	**44.5**	**13.4**	**13.6**	18.9	**33.8**	**54.8**	**42.7**	**77.8**	21.2	**58.2**
DMSR		56.8	37.6	8.7	13.4	**20.2**	25.0	42.1	35.0	66.5	**23.3**	48.2
Old-Net		33.2	9.5	0.3	0.7	2.6	4.1	14.4	11.1	44.2	3.8	19.5
LaPose	Frost	**68.0**	**41.6**	**13.4**	11.8	16.8	23.3	**49.7**	**40.0**	**76.9**	18.7	**52.9**
DMSR		55.6	35.3	8.0	**13.3**	**21.8**	**23.5**	40.6	33.5	66.2	**24.9**	45.6
Old-Net		35.6	12.0	0.3	0.4	1.2	4.8	13.1	14.0	47.6	1.9	17.0
LaPose	Fog	**70.3**	**47.1**	**16.2**	**15.3**	20.9	**36.2**	**57.4**	**45.8**	**78.8**	23.1	**60.4**
DMSR		55.0	34.6	7.5	14.2	**23.7**	23.1	43.8	31.1	65.1	**28.6**	50.9
Old-Net		17.3	2.3	0.1	0.1	0.3	1.1	4.8	3.9	27.7	0.6	7.3

Table 4. Ablation study of robustness with corrupted ROIs.

H. Per Category Results

Fig. 5 shows category-level pose estimation results using our method and each baseline which has public code [2, 10, 11]. Notably, our approach consistently improves the mean performance. Furthermore, our method outperforms others in the challenging non-symmetric camera and laptop categories. This result highlights the effectiveness of our object representation and the single-stage modeling strategy.

References

[1] Yamei Chen, Yan Di, Guangyao Zhai, Fabian Manhardt, Chenyangguang Zhang, Ruida Zhang, Federico Tombari, Nassir Navab, and Benjamin Busam. Secondpose: Se (3)-consistent dual-stream feature fusion for category-level pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9959–9969, 2024. 1

[2] Zhaoxin Fan, Zhenbo Song, Jian Xu, Zhicheng Wang, Kejian Wu, Hongyan Liu, and Jun He. Object level depth reconstruction for category level 6d object pose estimation from monocular rgb image. In *European Conference on Computer Vision*, pages 220–236. Springer, 2022. 3, 4, 7

[3] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018. 2

[4] Taeyeop Lee, Byeong-Uk Lee, Myungchul Kim, and In So Kweon. Category-level metric scale object shape and pose estimation. *IEEE Robotics and Automation Letters*, 6(4): 8575–8582, 2021. 3

[5] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 1

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[7] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. 2023. 1

[8] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 1, 2

[9] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 2, 5, 7

[10] 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 *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2036–2042. IEEE, 2024. 3, 4, 7

[11] Ruida Zhang, Ziqin Huang, Gu Wang, Chenyangguang Zhang, Yan Di, Xingxing Zuo, Jiwen Tang, and Xiangyang

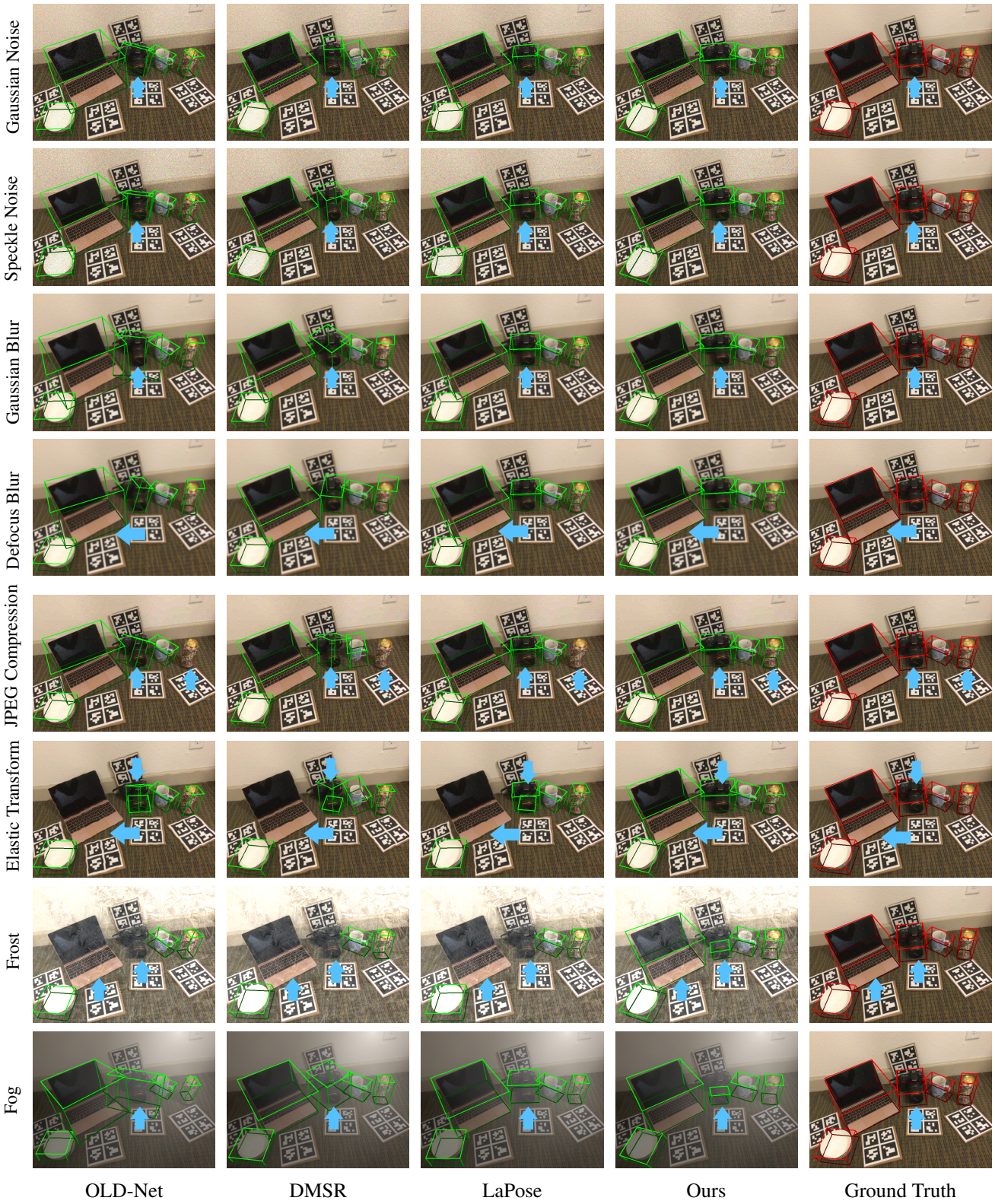

Figure 3. Qualitative comparison on Corrupted NOCS-REAL275[9]. We compare our model with all baselines (first to third columns) and with ground truth (last column) across 8 types of corruption.

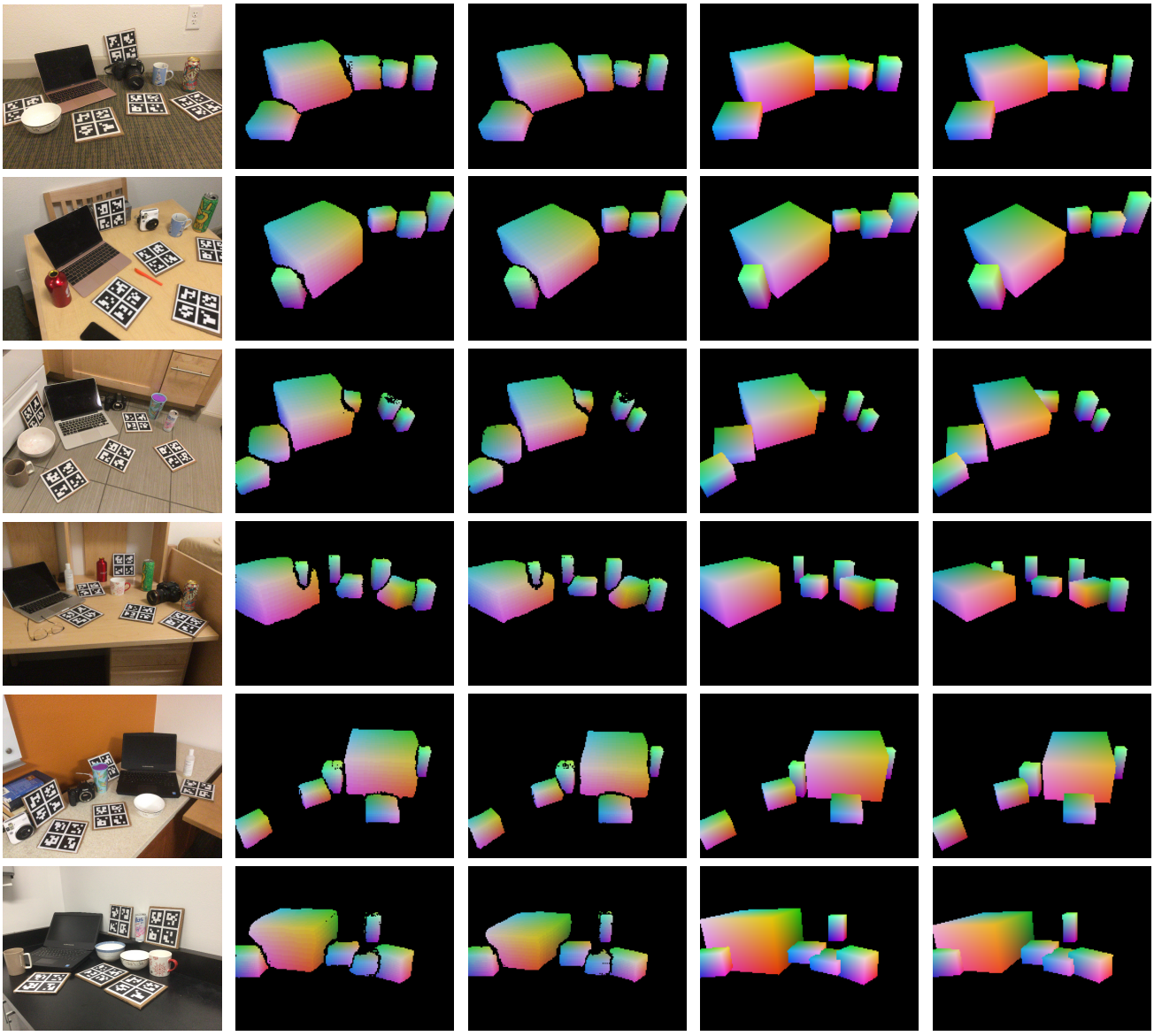

Figure 4. Visualization of the our dense matching results. We estimate 2D-3D correspondences between image features and our object prototypes with mean scales and instance-level scales. We remove low noisy correspondences using our foreground modeling strategy and confidence scores. Our method produces reliable and mostly noise-free correspondences in object regions.

Ji. Lapose: Laplacian mixture shape modeling for rgb-based category-level object pose estimation. In *European Conference on Computer Vision*, pages 467–484. Springer, 2024. 3, 4, 7

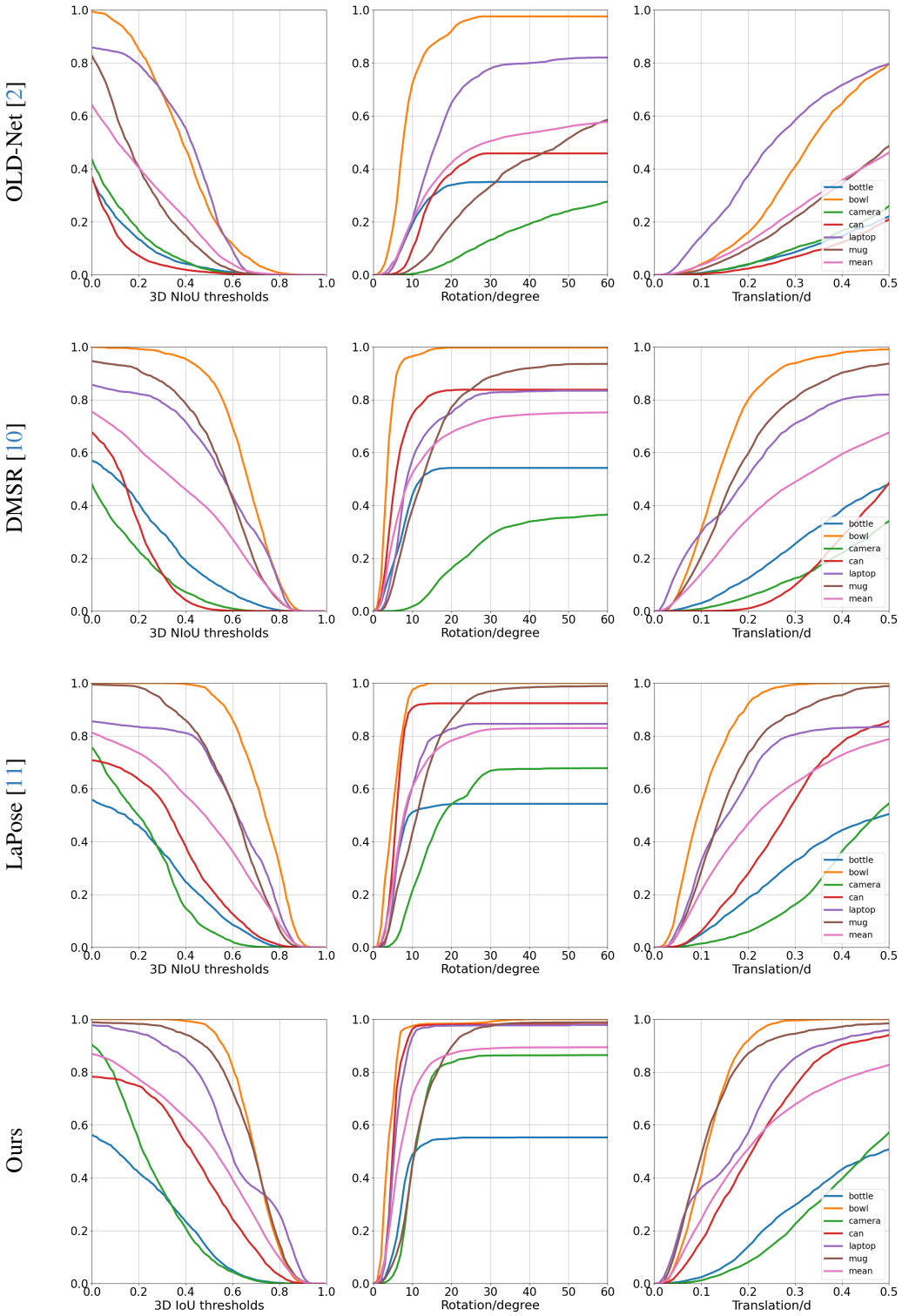

Figure 5. We show mean Average Precision (mAP) on REAL275[9] using scale-agnostic metrics. We compare our model with all baselines that have public code. Noticeably, our method has significantly increased rotation accuracy on the challenging non-symmetric categories camera and laptop.