Supplementary Material for the Paper: Towards Human-level 3D Relative Pose Estimation: Generalizable, Training-Free, with Single Reference

Yuan Gao*, Yajing Luo*, Junhong Wang, Kui Jia, Gui-Song Xia

We address the following issues in the supplementary material files:

1) Angle error distribution on LM-O in Sect. S1.
2) Experiments with imprecise input depth in Sect. S2.
3) Performance exploiting monocular *metric* depth estimation from advanced Depth Anything v2 [8] in Sect. S3.
4) Our results on the LineMOD [3], LM-O [1], and YCB-V [7] datasets w.r.t. **per object** in Sect. S4.
5) Qualitative Results on the LineMOD [3], LM-O [1], and YCB-V [7] datasets for all the methods in Sect. S5.
6) We also attached a video at https://www.youtube.com/watch?v=Ajr9ugjtoDo depicting the overview and the label/training-free refinement procedure of our method (i.e., the video version of Fig. 2 in the main text), as well as the qualitative results.

S1. Angle Error Distribution on the LM-O Dataset.

Figure S1 presents the angle error distribution (ranging from 0 to 180 degrees) for all the methods on the LM-O dataset. The statistics reveal that at lower angle error thresholds (e.g., for $t \leq 10, 20$ in Acc@$t°$), our approach substantially outperforms both 3DAHV [10] and DVMNet [9].

S2. Experiments with Imprecise Input Depth

As discussed in Sect. I.A (Applicability) in the main text, our method has the potential to use imprecise depth. We validate this on the LineMOD [3] dataset. Concretely, we simulate the imprecise depth by adding Gaussian noise $\mathcal{N}(0, \sigma)$ to the ground-truth depth map D_r, where σ is set to:

$$\sigma = \lambda * d, \qquad d = \max(D_r) - \min(D_r), \tag{1}$$

where d is the maximal depth difference of the input sample.

We validate different λ's with 0.001, 0.003, and 0.005 on the LineMOD [3] dataset, the results are shown in Table S1, which demonstrates that our method remains robust with imprecise depth obtained by a noisy depth sensor.

TABLE S1
Experiments with imprecise depth on LineMOD.

Metrics	Mean Err↓	Acc@30°↑	Acc@15°↑	Acc@10°↑	Acc@5°↑
$\lambda = 0.005$	33.10	69.52	52.66	40.16	20.64
$\lambda = 0.003$	30.92	71.44	54.96	42.90	23.10
$\lambda = 0.001$	30.11	72.00	55.10	43.04	24.22
Ours	**29.93**	**72.06**	**54.90**	**42.74**	**24.32**

S3. Performance exploiting monocular *metric* depth estimation

In order to further enhance our applicability without using reference depth as input, we further explored the advanced Depth Anything v2 (dpav2) [8] to estimate the monocular depth of our reference image.

To preserve the object shape, our method requires the ***metric*** depth, meaning the estimated depth z and spatial dimensions x, y should share the same unit of measurement (e.g., both in meters). This is in contrast to the ***relative*** depth, where the estimated depth and spatial appearance are subject to a scale ambiguity. Such an ambiguous scale distorts object shapes, e.g., given a centimeter spatial unit, an object could appear flattened if depth is measured in meters, or elongated if depth is in millimeters. Examples of scale-induced shape distortions are illustrated in Fig. S2.

In the following, we employ three configurations to obtain the metric depth from Depth Anything v2:

1) **Relative depth w/ Ground Truth align**: We use the relative depth estimated from the vanilla Depth Anything v2 model [8], and align its scale using the GT depth map. Denoted by *relative depth w/ GT align*, this approach yields the best results but is less practical as it requires GT depth annotations for scale alignment.
2) **In-dataset metric depth**: Following the procedure detailed in Section 7.3 of the Depth Anything v2 paper [8], we finetune the relative Depth Anything v2 model on the LineMOD [3] and YCB-V datasets [7], respectively, to obtain the corresponding metric depth models. Compared to *relative depth w/ GT align*, *in-dataset metric depth* is more practical but provides inferior results.
3) **Cross-dataset metric depth**: For zero-shot cross-dataset testing in our experiments, we employ the metric depth estimation model provided by Depth Anything v2, which was pretrained on the external Hypersim dataset [5]. This configuration *fully eliminates the requirement for in-dataset depth finetuning*. However, it yields limited accuracy, likely due to an imprecisely recovered depth

Y. Gao and G.-S. Xia are with the School of Artificial Intelligence, Wuhan University, Wuhan, China. E-mails: ethan.y.gao@gmail.com, guisong.xia@whu.edu.cn
Y. Luo is with the School of Computer Science, Wuhan University, Wuhan, China. E-mail: yajingluo@whu.edu.cn
J. Wang is with MoreFun Studio, Tencent Games, Tencent, Shenzhen, China. E-mail: junhongwang@tencent.com
K. Jia is with the School of Data Science, The Chinese University of Hong Kong, Shenzhen, China. E-mail: kuijia@cuhk.edu.cn
Corresponding authors: Yuan Gao, Gui-Song Xia.
* indicates equal contributions.

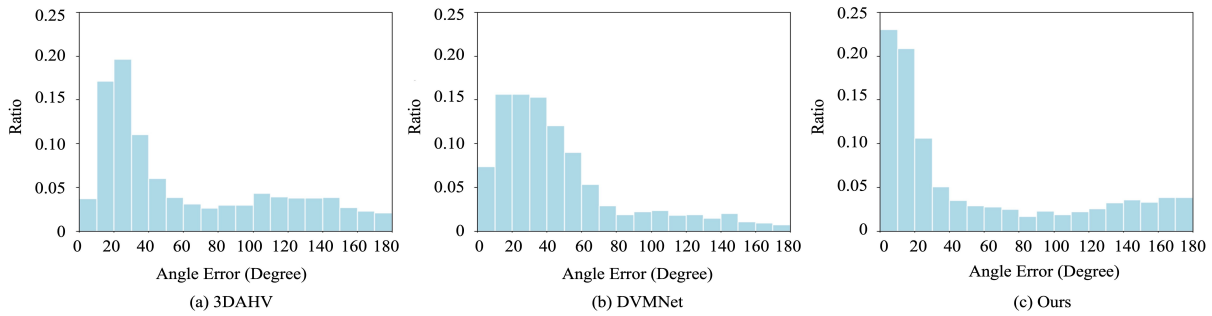

Fig. S1. **Angle Error Distribution on LM-O.**

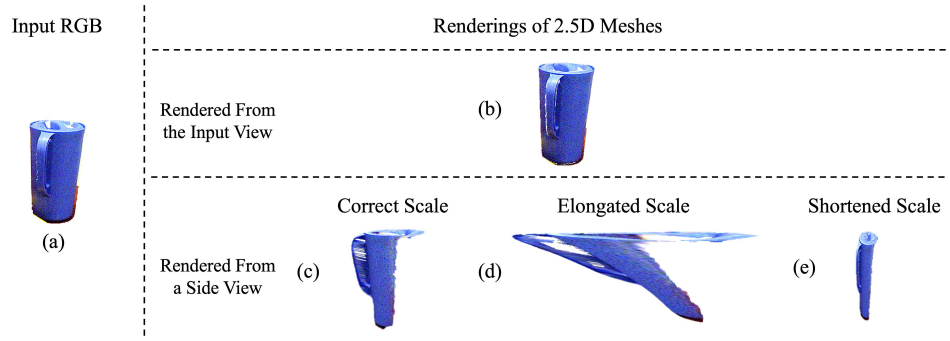

Fig. S2. **Illustration of scale-induced shape distortions.** Given an input image (a), the estimated depth z should align with the spatial dimensions x, y in scale, so as to obtain the correct shape (c). Incorrect depth scales will result in an elongated shape (d) or shortened shape (e). Subfigures (c), (d), and (e) are from a rotated side view of (b) for clearer illustrations. We thus require the ***metric*** depth **with a correct depth scale**, rather than the ***relative*** depth, to preserve the object shape.

scale caused by varying camera parameters and/or objects between the Hypersim pretraining set and zero-shot testing sets LineMOD [3], LM-O [1], and YCB-V [7].

The results shown in Tables S2 - S4 demonstrate that:

1) Ours, Ours (dpav2, relative depth w/ GT align), and Ours (dpav2, in-dataset metric depth) all outperform both SOTA DVMNet (in-dataset) and DVMNet (cross-dataset) [9] across the rigorous Acc $@5°, 10°, 15°$.
2) Our most applicable configuration, which is training-free and requires neither depth nor pose annotations, i.e., Ours (dpav2, cross-dataset metric depth), outperforms the SOTA DVMNet (cross-dataset) on the LineMOD dataset. However, it is inferior on YCB-V and LM-O, likely because that the heavy occlusions in these datasets lead to worse cross-dataset metric depth estimation.

Note that both the *relative depth w/ GT align* and *in-dataset metric finetune* approaches require additional depth labels to align or finetune a metric depth estimation model. On the other hand, the *cross-dataset metric depth* configuration performs suboptimally on some benchmarking datasets. Our current requirement for the reference depth is likely to be alleviated once a generalizable metric depth estimator becomes available.

TABLE S2
ABLATION OF PREDICTED DEPTH ON LINEMOD.

Method	Error↓	Acc @ $t°$ (%) ↑			
	Mean Err	30°	15°	10°	5°
DVMNet (cross-dataset)	47.47	36.44	13.14	5.92	1.08
DVMNet (in-dataset)	33.28	55.02	22.38	10.66	2.72
Ours (dpav2, relative depth w/ GT align)	32.04	70.34	48.98	34.84	15.32
Ours (dpav2, in-dataset metric depth)	41.54	59.80	34.18	21.7	8.38
Ours (dpav2, cross-dataset metric depth)	54.04	43.22	20.02	11.24	3.52
Ours	**29.93**	**72.06**	**54.90**	**42.74**	**24.32**

TABLE S3
ABLATION OF PREDICTED DEPTH ON LM-O.

Method	Error↓	Acc @$t°$ (%) ↑			
	Mean Err	30°	15°	10°	5°
DVMNet (cross-dataset)	51.75	35.52	12.94	5.30	1.33
DVMNet (in-dataset)	**48.55**	38.62	14.14	7.37	1.87
Ours (dpav2, relative depth w/ GT align)	59.28	47.70	28.78	18.31	5.50
Ours (dpav2, in-dataset metric depth)	66.58	41.56	22.58	13.11	2.53
Ours (dpav2, cross-dataset metric depth)	75.14	29.36	11.70	5.25	1.04
Ours	55.09	**54.50**	**34.97**	**23.00**	**6.83**

TABLE S4
ABLATION OF PREDICTED DEPTH ON YCB-V.

Method	Error↓	Acc @ $t°$ (%) ↑			
	Mean Err	30°	15°	10°	5°
DVMNet (cross-dataset)	54.12	41.28	17.11	9.35	2.53
DVMNet (in-dataset)	48.88	51.71	27.04	14.03	3.16
Ours (dpav2, relative depth w/ GT align)	54.54	53.21	40.19	29.44	13.41
Ours (dpav2, in-dataset metric depth)	57.15	49.10	33.71	23.64	10.05
Ours (dpav2, cross-dataset metric depth)	69.90	28.71	10.98	6.06	1.89
Ours	**47.09**	**56.63**	**42.69**	**31.86**	**14.18**

TABLE S5
OUR RESULTS ON THE LINEMOD DATASET W.R.T. PER OBJECT. THE OBJECTS WITH RED TEXT ARE THOSE USED FOR TESTING IN THE MAIN PAPER.

Object	Mean Err↓	Acc@30°↑	Acc@15°↑	Acc@10°↑	Acc@5°↑
ape	43.41	46.90	26.10	17.40	5.80
benchvise	17.79	87.30	75.60	64.60	42.10
camera	24.10	73.70	58.00	46.80	27.60
can	26.49	75.00	63.20	55.00	37.80
cat	33.90	68.00	52.20	40.20	22.00
driller	35.58	76.90	59.40	44.60	23.90
duck	38.30	54.40	29.30	17.50	6.00
eggbox	27.63	77.40	66.10	57.10	36.90
glue	46.35	55.30	38.20	28.00	13.70
holepuncher	26.25	76.50	64.30	52.30	26.80
iron	33.20	73.70	58.80	48.90	29.60
lamp	29.28	79.30	66.20	56.20	36.90
phone	25.82	80.80	64.60	50.20	27.30
average	31.39	71.17	55.54	44.52	25.88

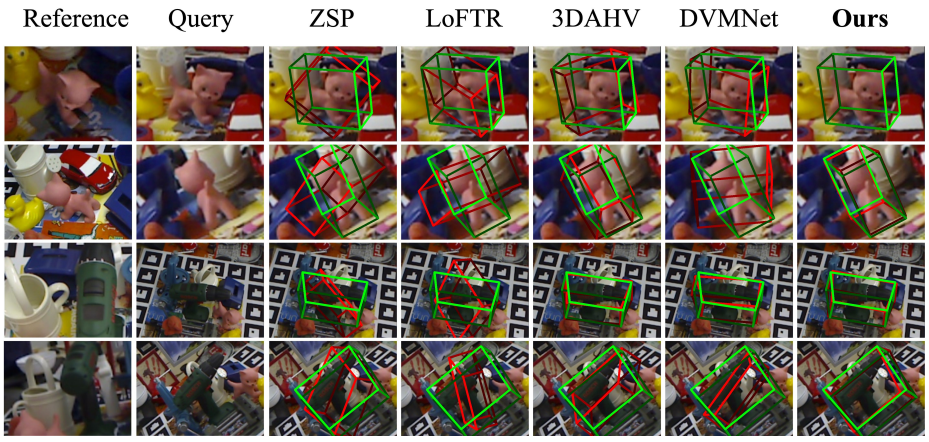

Fig. S3. **Qualitative results on LineMOD.** RelPose++ [4] did not release the LineMOD weights, the results of it in our main text were pasted from the 3DAHV paper [10], therefore the visualized results of RelPose++ are not included.

TABLE S6
OUR RESULTS ON THE LM-O DATASET W.R.T. PER OBJECT. THE OBJECTS WITH RED TEXT ARE THOSE USED FOR TESTING IN THE MAIN PAPER.

Object	Mean Err↓	Acc@30°↑	Acc@15°↑	Acc@10°↑	Acc@5°↑
ape	60.85	42.10	22.40	11.60	2.10
can	38.49	63.60	53.10	41.70	19.60
cat	55.84	53.70	33.00	22.30	8.60
driller	52.29	59.10	42.20	28.50	7.30
duck	57.13	50.70	29.70	18.20	4.60
eggbox	45.94	58.90	43.40	34.50	16.00
glue	61.52	46.50	30.70	19.20	9.20
holepuncher	42.51	62.20	47.20	33.00	10.40
average	51.82	54.59	37.72	26.13	9.73

S4. PER OBJECT RESULTS ON THE LINEMOD, LM-O, AND YCB-V DATASETS

We present our results w.r.t. per object of the full LineMOD [3], LM-O [1], and YCB-V [7] datasets in Tables S5, S6, and S7, respectively. The experimental settings are the same as those in the main text, i.e., Tables II-IV.

Tables S5, S6, and S7 show that our method performs well on all the objects of the three datasets without training, further validating the strong zero-shot unseen-object generalize-ability of our label/training-free method.

S5. QUALITATIVE RESULTS ON THE LINEMOD, LM-O AND YCB-V DATASETS

Qualitative results on the LineMOD [3], LM-O [1], and YCB-V [7] datasets are illustrated in Figs. S3, S4, and S5, respectively. The ground truth and predicted poses are visualized by axes and 3D bounding boxes.

As depicted in Figs. S3, S4, and S5, our method outperforms the state-of-the-art methods [2, 4, 6, 9, 10] qualitatively in all the three datasets.

REFERENCES

[1] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *ECCV*, pages 536–551. Springer, 2014. 1, 2, 3

[2] Walter Goodwin, Sagar Vaze, Ioannis Havoutis, and Ingmar Posner. Zero-shot category-level object pose estimation. In *ECCV*, pages 516–532. Springer, 2022. 3

[3] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *ACCV*, pages 548–562. Springer, 2012. 1, 2, 3

[4] Amy Lin, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. RelPose++: Recovering 6d poses from sparse-view observations. *arXiv preprint arXiv:2305.04926*, 2023. 3, 4

[5] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, pages 10912–10922, 2021. 1

[6] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *CVPR*, pages 8922–8931, 2021. 3

[7] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 1, 2, 3

[8] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything v2. In *NeurIPS*, volume 37, pages 21875–21911, 2024. 1

[9] Chen Zhao, Tong Zhang, Zheng Dang, and Mathieu Salzmann. DVMNet: Computing relative pose for unseen objects beyond hypotheses. In *CVPR*, pages 20485–20495, 2024. 1, 2, 3

[10] Chen Zhao, Tong Zhang, and Mathieu Salzmann. 3d-aware hypothesis & verification for generalizable relative object pose estimation. In *ICLR*, 2024. 1, 3

TABLE S7
OUR RESULTS ON THE YCB-V DATASET W.R.T. PER OBJECT. THE OBJECTS WITH RED TEXT ARE THOSE USED FOR TESTING IN THE MAIN PAPER.

Object	Mean Err↓	Acc@30°↑	Acc@15°↑	Acc@10°↑	Acc@5°↑
002_master_chef_can	59.81	42.40	31.40	19.80	9.40
003_cracker_box	83.20	40.40	32.70	26.80	8.50
004_sugar_box	44.93	68.00	62.30	55.70	30.10
005_tomato_soup_can	61.65	36.50	25.80	19.70	9.40
006_mustard_bottle	20.76	86.40	83.30	77.80	59.70
007_tuna_fish_can	111.75	16.50	10.20	8.10	4.70
008_pudding_box	11.14	95.20	74.30	63.80	33.90
009_gelatin_box	8.13	99.50	87.20	78.30	32.40
010_potted_meat_can	99.01	26.80	22.30	13.20	4.60
011_banana	46.94	56.40	46.00	33.60	12.30
019_pitcher_base	33.43	58.60	41.40	28.50	9.50
021_bleach_cleanser	51.91	55.80	41.00	29.70	12.00
024_bowl	17.89	90.50	63.80	35.20	11.10
025_mug	43.98	40.80	22.90	15.50	3.30
035_power_drill	42.45	67.60	48.90	32.50	13.10
036_wood_block	39.52	69.10	48.50	30.50	10.50
037_scissors	27.55	83.50	60.00	41.10	15.40
040_large_marker	52.96	25.40	10.00	5.00	1.80
051_large_clamp	71.20	43.10	26.60	18.80	7.70
052_extra_large_clamp	88.61	27.00	14.90	7.40	2.80
061_foam_brick	27.89	81.90	73.00	50.10	13.30
average	49.74	57.69	44.12	32.91	14.55

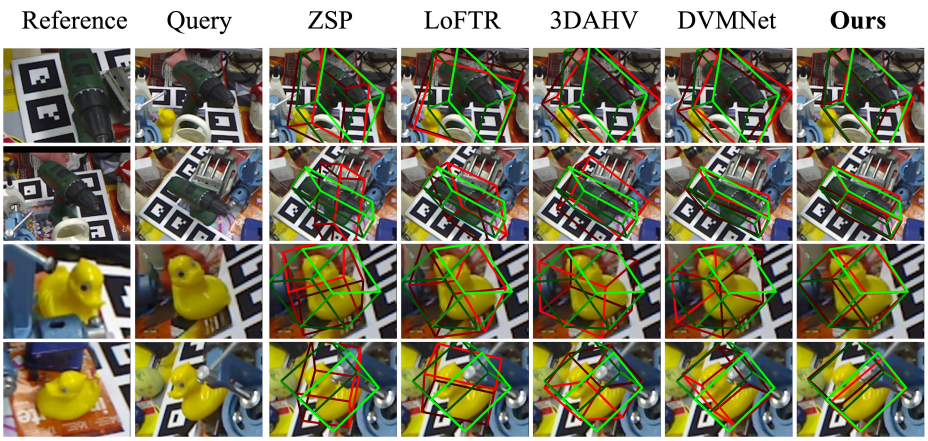

Fig. S4. **Qualitative results on LM-O.** LM-O is typically used solely to evaluate the models trained on LineMOD, since RelPose++ [4] have not released the LineMOD weights, the visualized results of RelPose++ are not included.

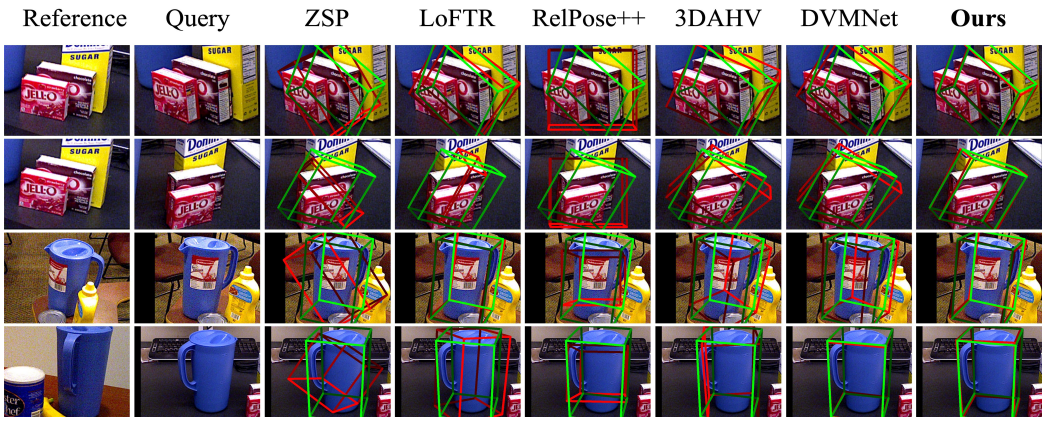

Fig. S5. **Qualitative results on YCB-V.** The predicted poses are visualized by red 3D bounding boxes while the ground truth poses are depicted by green 3D bounding boxes.