Knowledge Distillation for 6D Pose Estimation by Aligning Distributions of Local Predictions - Supplementary Material -

Shuxuan Guo[1], Yinlin Hu[2], Jose M. Alvarez[3], Mathieu Salzmann[1,4]
[1]CVLab, EPFL [2]MagicLeap [3]NVIDIA [4]ClearSpace
shuxuan.guo@epfl.ch yhu@magicleap.com josea@nvidia.com mathieu.salzmann@epfl.ch

S1. Code for our Approach

Our code can be found at: https://github.com/GUOShuxuan/kd-6d-pose-adlp. The details of how to build up the environment and run our main experiments are in the README.md file.

Below, we provide the details of the existing assets we used in our work, such as the LINEMOD [7], Occluded-LINEMOD [1] and YCB-V [12] datasets, the GeomLoss library [5] and the original WDRNet [9] and ZebraPose [11] codebase. All of them are open source and available for non-commercial academic research.

LINEMOD [7] and Occluded-LINEMOD [1][1] are 6D pose estimation benchmarks, consisting of 3D object models, training and test RGB/RGB-D images annotated with ground-truth 6D object poses and intrinsic camera parameters. In our work, we do not use the RGB-D data. The LINEMOD dataset consists of 15 texture-less household objects with discriminative color, shape and size. Only 13 of the objects have the CAD models, so, following standard practice, we focus on them. Each object is associated with a training/testing image set showing one annotated object instance with significant clutter but only mild occlusion. Following [2], we split the data into a training set containing around 200 images per object and a test set containing around 1000 images per object. Occluded-LINEMOD provides additional ground-truth annotations for all modeled objects in one of the test sets from LINEMOD. This introduces challenging test cases with various levels of occlusion. Note that we use the real images from LINEMOD together with the synthetic ones provided with the dataset and generated using physically-based rendering [8]. In our work, we respect the terms and conditions of use listed on the websites.

YCB-V [12][2] is a large-scale video dataset for 6D object pose estimation, which provides accurate 6D poses of 21 objects observed in 92 videos, with in total of 133,827 frames. The objects are highly occluded. For the training of YCB-V, we make use of both the 110k+ real images and the public synthetic data using physically-based rendering (pbr) [8].

WDRNet [9][3] and **ZebraPose** [11][4] are open-source 6D pose estimation frameworks built in Pytorch [10], and are released under the non-commercial use license and MIT License, respectively. Together with WDRNet, we also exploit the detector pre-processing portion of the SO-Pose [4] codebase[5], which is released under the Apache License 2.0. To implement and solve the Optimal Transport (OT) models in our method, we rely on the GeomLoss library [5][6], which falls under the MIT License. For the details of these licenses, please refer to the websites.

Computing resources. All experiments were conducted on an internal cluster, with Tesla V100 or A100 GPUs. All models were trained on one single GPU.

S2. Hyper-parameters for Naive-KD and FKD

In this section, as mentioned in the main paper, we provide the details of the hyper-parameter search for Naive-KD and FKD [13]. In both cases, this search was mostly focused on models with a DarkNet-tiny-H backbone and on 2 difficult LINEMOD classes, i.e., Ape and Duck.

Naive-KD. In the sparse 2D keypoints scenario, for WDRNet+, as shown in Table S1, the best results are obtained with a norm $p = 1$ and a distillation loss weight of 0.1, and with a norm $p = 2$ with a weight of 0.1. We therefore provide the corresponding results for all classes and for the DarkNet-tiny-H and DarkNet-tiny backbones in Table S2. Note that $p = 2$ with a weight of 0.1 yields the best results for DarkNet-tiny-H, and $p = 1$ with a weight of 0.1

[1] https://bop.felk.cvut.cz/datasets
[2] https://rse-lab.cs.washington.edu/projects/posecnn
[3] https://github.com/cvlab-epfl/wide-depth-range-pose
[4] https://github.com/suyz526/ZebraPose
[5] https://github.com/shangbuhuan13/SO-Pose
[6] https://github.com/jeanfeydy/geomloss

Table S1. **Results of Naive-KD with DarkNet-tiny-H backbone on Ape and Duck with WDRNet+.** We report the ADD-0.1d for the Naive-KD with $p = 1$ and $p = 2$.

Class	Teacher	Student	$p = 1$			$p = 2$		
			0.01	**0.1**	1.0	0.01	**0.1**	1.0
Ape	82.6	65.4	63.2	64.4	65.7	63.8	64.1	64.8
Duck	76.0	64.3	59.4	63.3	60.3	59.0	63.6	62.2
AVG.	79.3	64.8	61.3	**63.9**	63.0	61.4	**63.9**	63.5

Table S2. **Results of Naive-KD on LINEMOD dataset with WDRNet+.** We report the ADD-0.1d for the Naive-KD with DarkNet-tiny-H and DarkNet-tiny backbones with the different norms p and the weights searched from Table S1.

Class	Teacher	DarkNet-tiny-H			DarkNet-tiny		
		Student	$p = 1$ 0.1	$p = 2$ **0.1**	Student	$p = 1$ **0.1**	$p = 2$ 0.1
Ape	82.6	65.4	64.4	64.1	73.4	74.1	74.0
Bvise	95.5	92.0	90.6	91.4	95.2	95.4	96.6
Cam	93.8	78.4	77.8	79.1	91.2	89.7	90.0
Can	95.7	82.2	78.7	81.0	94.4	92.7	92.9
Cat	92.0	81.5	77.8	78.7	87.2	85.0	82.0
Driller	94.8	85.5	87.6	87.4	92.2	93.1	93.2
Duck	76.0	64.3	63.3	63.6	70.9	74.4	73.9
Eggbox*	99.1	95.8	95.3	95.0	99.3	98.7	99.4
Glue*	96.4	90.7	92.6	91.2	97.2	97.1	96.9
Holep	86.2	73.2	71.6	72.3	78.0	82.1	81.0
Iron	93.6	86.3	86.4	86.3	92.1	92.1	91.9
Lamp	97.7	93.6	93.3	94.2	96.6	95.3	96.5
Phone	91.2	76.0	75.7	75.8	87.5	88.4	87.4
AVG.	91.9	81.9	81.2	**81.6**	88.9	**89.1**	88.9

Table S3. **Weight searching for Naive-KD on OCC-LINEMOD dataset with ZebraPose (Ape and Duck).** We report the ADD-0.1d for Naive-KD with different weights.

Class	Teacher	Student	0.1	**1.0**	10.0
Ape	57.9	47.2	49.1	51.1	50.5
Duck	64.5	57.2	57.4	60.7	59.7
AVG.	61.2	52.2	53.3	**55.9**	55.1

gets the best performance for DarkNet-tiny. Therefore, we report the best results for each backbone in the main paper. Note that, for WDRNet+, Naive-KD hardly improves the student's performance.

For ZebraPose, we use $p = 1$ for the DarkNet-tiny student backbone, with a weight in $\{0.1,1,10\}$. As shown in Table S3, a weight of 1.0 yields the best results.

Table S4. **Weight searching for FKD on LINEMOD dataset with WDRNet+ (Ape and Duck).** We report the ADD-0.1d for FKD [13] with different weights.

Class	Teacher	Student	0.001	**0.01**	0.1	1.0
Ape	82.6	65.4	66.5	68.4	66.5	65.0
Duck	76.0	64.3	65.2	66.8	61.2	60.3
AVG.	79.3	64.8	65.9	**67.6**	63.8	62.7

Table S5. **Results of FKD on OCC-LINEMOD dataset with WDRNet+.** We report the ADD-0.1d for FKD [13] with different weights. Note that due to the worse results on Ape and Duck with a weight of 0.1, we didn't extend this setting to other classes.

Class	Teacher	Student	0.001	**0.01**	0.1
Ape	33.4	25.5	26.8	26.7	22.6
Can	70.9	46.6	52.8	53.9	-
Cat	45.1	31.4	31.0	31.1	-
Driller	70.9	51.2	52.3	52.1	-
Duck	27.0	22.5	24.7	25.3	19.8
Eggbox*	53.7	43.4	47.9	49.0	-
Glue*	70.7	54.5	54.3	55.6	-
Holep	59.7	49.3	51.0	52.2	-
AVG.	53.9	40.5	42.6	**43.2**	-

FKD [13]. We follow the same strategy as above, and report the results for Ape and Duck with FKD in Table S4. The best results are obtained with a distillation weight of 0.01. As the weight increases, the performance decreases significantly. We therefore adopted 0.01 as FKD weight for both the DarkNet-tiny-H and DarkNet-tiny backbones on the LINEMOD dataset. For FKD, we also conducted a hyper-parameter search on Occluded-LINEMOD. As shown in Table S5, a distillation weight of 0.01 also achieves the best results. Note that we did not test a weight of 0.1 on all classes because of the worse results it gave on Ape and Duck.

S3. Hyper-parameters for our Approach

In this section, we include the hyper-parameter search for our proposed keypoint distribution alignment distillation method, including the norm p and the weight of our distillation loss. As for WDRNet+, we focused this search on DarkNet-tiny-H for the Ape and Duck classes. As shown in Table S6, $p = 2$ yields much better results than $p = 1$, and we therefore use $p = 2$ in the main paper. As for the loss weight, on the LINEMOD dataset, 5 yields the best results, which we use to report the results on the 13 classes in the main paper. For Occluded-LINEMOD, as shown in Table S7, we obtain the best results with a weight of 0.1. Note that our preliminary experiments with a weight of 1 showed worse performance, and we thus did not compute full results with weights larger than 0.1.

For ZebraPose, as shown in Table S8, we observed a weight of 1.0 to only yield a marginal improvement on the class Ape. We therefore increased the weight to 10.0 and 100.0, both of which led to higher improvements on Ape. These improvements also materialized on the other classes. Thus, in the main paper, we report the results with a weight

Table S6. **Results of our proposed KD with DarkNet-tiny-H backbone on LINEMOD dataset (Ape and Duck) with WDR-Net+.** We report the ADD-0.1d for our proposed KD with different ps and weights.

Class	Teacher	Student	$p = 1$		$p = 2$		
			1.0	10.0	1.0	**5.0**	10.0
Ape	82.6	65.4	61.9	61.5	66.5	69.4	67.0
Duck	76.0	64.3	61.2	61.9	65.1	66.5	65.8
AVG.	79.3	64.8	61.6	61.7	65.8	**67.9**	66.4

Table S7. **Results of our proposed KD on OCC-LINEMOD dataset with WDRNet+.** We report the ADD-0.1d for our proposed KD with different weights.

Class	Teacher	Student	0.01	**0.1**
Ape	33.4	25.5	23.5	25.7
Can	70.9	46.6	51.2	53.5
Cat	45.1	31.4	31.3	32.2
Driller	70.9	51.2	51.5	52.9
Duck	27.0	22.5	20.0	25.7
Eggbox*	53.7	43.4	47.9	48.2
Glue*	70.7	54.5	54.3	55.8
Holep	59.7	49.3	51.0	52.1
AVG.	53.9	40.5	41.3	**43.2**

Table S8. **Results of our proposed KD on OCC-LINEMOD dataset with ZebraPose.** We report the ADD-0.1d for our proposed KD with different weights.

Class	Teacher	Student	1.0	10.0	**100.0**
Ape	57.9	47.2	47.9	50.1	**52.0**
Can	95.0	93.2	-	94.0	**94.2**
Cat	60.6	53.1	-	54.8	**55.2**
Driller	94.8	90.3	-	89.1	**90.4**
Duck	64.5	57.2	-	60.8	**61.0**
Eggbox*	70.9	69.6	-	70.4	**70.7**
Glue*	88.7	84.1	-	84.3	**84.3**
Holep	83.0	75.8	-	76.9	**78.8**
AVG.	76.9	71.4	-	72.5	**73.3**

of 100.0.

S4. Comparison with lightweight networks on 6D pose estimation

We note that HRPose [6], CRT-6D [3] and FFN [14] also work on the lightweight networks on 6D pose estimation task. Therefore, we compare these works with ours by providing the input types, networks, #Params and #GFLOPS in Table S9. CRT-6D consists of a ReNet34 backbone with a pose refinement transformer module. It is thus much larger than our students and it estimates the pose in 36ms vs 20ms for our students. FFN takes a depth map as additional input, while we focus on the RGB-only case. Moreover, with 24.5M parameters, it is much larger than our students. Moreover, for comparison, we replaced the WDRNet+ backbone with the small HRNetV2-W18 (as in HRPose), and achieved 88.17, outperforming HRPose (87.55), and requiring fewer parameters and GFLOPs thanks to a lighter regression head.

Table S9. **Comparison of different lightweight models.**

Model	Input	Network	#Params(M)	#GFLOPs
CRT-6D	RGB-only	ResNet34 + Transformer	> 21.8	-
FFN	*Depth* + RGB	MobileNetV2 + PSPNet	24.5	-
HRPose	RGB-only	small HRNetV2-W18	4.2	15.5
Our Student (WDRNet+)	RGB-only	small HRNetV2-W18 DarkNet-tiny-H DarkNet-tiny	4.1 2.3 8.5	4.6 4.8 17.3

S5. Additional Qualitative Analysis

In this section, we showcase some failure cases with our distilled student models on several examples from Occluded-LINEMOD. As shown in Figure S1, our main failure cases arise from poor teacher predictions, which are unable to improve the student training. Exploring better ways to leverage the teacher's knowledge would be an interesting research topic in our future work.

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 *European Conference on Computer Vision*, 2014. 1

[2] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image. In *Conference on Computer Vision and Pattern Recognition*, 2016. 1

[3] Pedro Castro and Tae-Kyun Kim. CRT-6D: Fast 6D Object Pose Estimation with Cascaded Refinement Transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5746–5755, 2023. 3

[4] Yan Di, Fabian Manhardt, Gu Wang, , Xiangyang Ji, Nassir Navab, and Federico Tombari. SO-Pose: Exploiting Self-Occlusion for Direct 6D Pose Estimation. In *International Conference on Computer Vision*, 2021. 1

[5] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve, and Gabriel Peyré. Interpolating between Optimal Transport and MMD using Sinkhorn Divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019. 1

[6] Qi Guan, Zihao Sheng, and Shibei Xue. HRPose: Real-Time High-Resolution 6D Pose Estimation Network Using Knowledge Distillation. *Chinese Journal of Electronics*, 32(1):189–198, 2023. 3

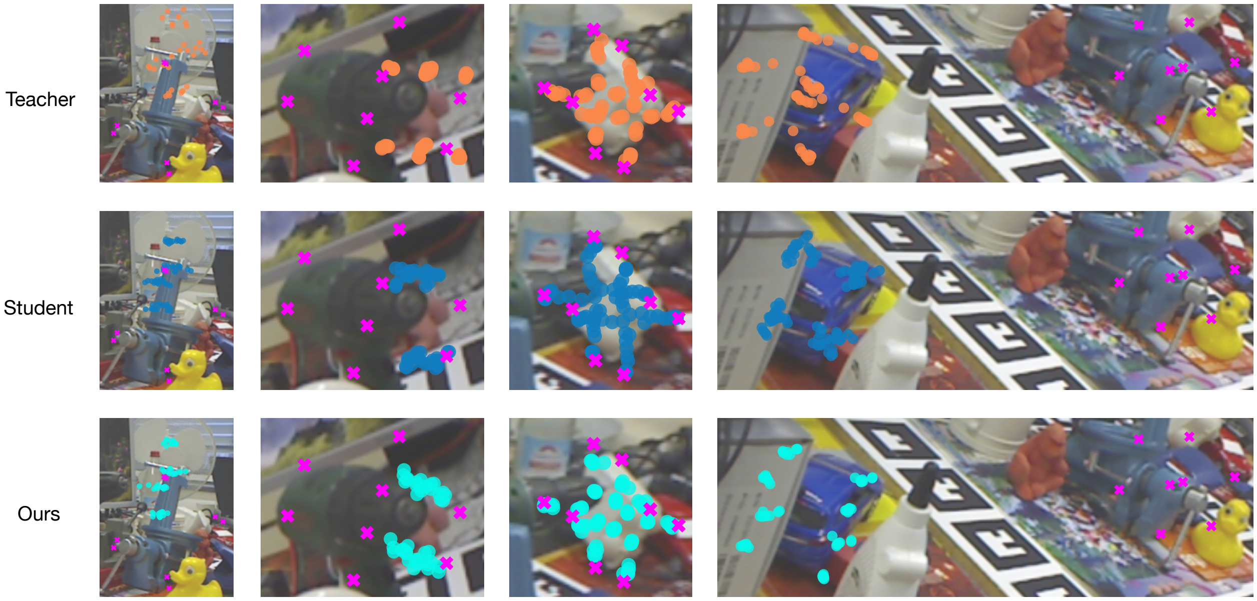

Figure S1. **Qualitative Analysis of Failure Cases from Occluded-LINEMOD dataset** (better viewed in color). Comparison of the 2D keypoints predicted with the teacher model (1st row, orange dots), the baseline student model (2nd row, dark blue dots) and our distilled student model (last row, light blue dots). Our distilled student model is able to mimic the teacher's keypoints distribution, however, poor teacher predictions can negatively impact the distilled student predictions.

[7] 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 *Asian Conference on Computer Vision*, 2012. 1

[8] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. BOP Challenge 2020 on 6D Object Localization. *European Conference on Computer Vision Workshops*, 2020. 1

[9] Yinlin Hu, Sébastien Speierer, Wenzel Jakob, Pascal Fua, and Mathieu Salzmann. Wide-Depth-Range 6D Object Pose Estimation in Space. In *Conference on Computer Vision and Pattern Recognition*, 2021. 1

[10] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. 2017. 1

[11] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. ZebraPose: Coarse to Fine Surface Encoding for 6DoF Object Pose Estimation. *Conference on Computer Vision and Pattern Recognition*, 2022. 1

[12] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. *arXiv Preprint*, 2017. 1

[13] Linfeng Zhang and Kaisheng Ma. Improve Object Detection with Feature-based Knowledge Distillation: Towards Accurate and Efficient Detectors. In *International Conference on Learning Representations*, 2021. 1, 2

[14] Ligang Zuo, Lun Xie, Hang Pan, and Zhiliang Wang. A Lightweight Two-End Feature Fusion Network for Object 6D Pose Estimation. *Machines*, 2022. 3