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

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

Abstract

Knowledge distillation facilitates the training of a compact student network by using a deep teacher one. While this has achieved great success in many tasks, it remains completely unstudied for image-based 6D object pose estimation. In this work, we introduce the first knowledge distillation method driven by the 6D pose estimation task. To this end, we observe that most modern 6D pose estimation frameworks output local predictions, such as sparse 2D keypoints or dense representations, and that the compact student network typically struggles to predict such local quantities precisely. Therefore, instead of imposing prediction-to-prediction supervision from the teacher to the student, we propose to distill the teacher's distribution of local predictions into the student network, facilitating its training. Our experiments on several benchmarks show that our distillation method yields state-of-the-art results with different compact student models and for both keypoint-based and dense prediction-based architectures.

1. Introduction

Estimating the 3D position and 3D orientation, a.k.a. 6D pose, of an object relative to the camera from a single 2D image has a longstanding history in computer vision, with many real-world applications, such as robotics, autonomous navigation, and virtual and augmented reality. Modern methods that tackle this task [7, 20, 21, 25, 28, 33, 40, 45, 47] all rely on deep neural networks. The vast majority of them draw their inspiration from the traditional approach, which consists of establishing correspondences between the object's 3D model and the input image and compute the 6D pose from these correspondences using a Perspective-n-Point (PnP) algorithm [2, 23, 27, 42] or a learnable PnP network. Their main differences then lie in the way they extract correspondences. While some methods predict the 2D image locations of sparse 3D object keypoints, such as the 8 3D bounding box corners [19–21] or points on the object surface [33], others produce dense representations, such as 3D locations [7, 45] or binary codes [40], from which the pose can be obtained.

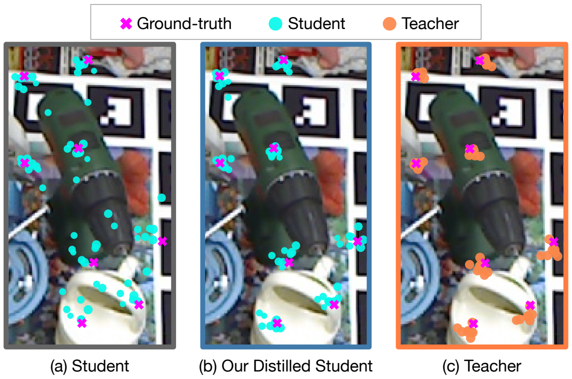

Figure 1. **Student vs teacher keypoint predictions.** The large backbone of the teacher allows it to produce accurate keypoints, indicated by tight clusters. By contrast, because of its more compact backbone, the student struggles to predict accurate keypoints when trained with keypoint-to-keypoint supervision. We therefore propose to align the student's and teacher's keypoint *distributions*.

In any event, these methods rely on large models, which, while achieving impressive accuracy, are impractical deployment on embedded platforms and edge devices. As, to the best of our knowledge, no compact and efficient 6D pose estimation models have yet been proposed, a simple way to reduce the size of these networks consists of replacing their large backbones with much smaller ones. Unfortunately, this typically comes with a significant accuracy drop. In this paper, we address this by introducing a knowledge distillation strategy for 6D pose estimation networks.

Knowledge distillation aims to transfer information from a deep teacher network to a compact student one. The research on this topic has tackled diverse tasks, such as image classification [17, 37, 48], object detection [10, 11, 49] and semantic segmentation [14, 30]. While some techniques, such as feature distillation [15, 37, 48, 49], can in principle generalize to other tasks, no prior work has studied knowledge distillation in the context of 6D pose estimation.

In this paper, we introduce a knowledge distillation method for 6D pose estimation motivated by the following observations. In essence, whether outputting sparse 2D locations or dense representations, the methods discussed above all produce multiple local predictions. We then argue that the main difference between the local predictions made by a deep teacher network and a compact student one consists in the accuracy of these individual predictions. Figure 1 showcases this for sparse keypoint predictions, evidencing that predicting accurate keypoint locations with keypoint-to-keypoint supervision is much harder for the student than for the teacher. We therefore argue that knowledge distillation for 6D pose estimation should be performed not by matching the individual local predictions of the student and teacher but instead by encouraging the student and teacher *distributions* of local predictions to become similar. This leaves more flexibility to the student and thus facilitates its training.

To achieve this, we follow an Optimal Transport (OT) formalism [44], which lets us measure the distance between the two sets of local predictions. We express this as a loss function that can be minimized using a weight-based variant of Sinkhorn's algorithm [6], which further allows us to exploit predicted object segmentation scores in the distillation process. Our strategy is invariant to the order and the number of local predictions, making it applicable to unbalanced teacher and student predictions that are not in one-to-one correspondence.

We validate the effectiveness of our approach by conducting extensive experiments on the popular LINEMOD [16], Occluded-LINEMOD [3] and YCB-V [47] datasets with the SOTA keypoint-based approach WDRNet+. Our prediction distribution alignment strategy consistently outperforms both a prediction-to-prediction distillation baseline and the state-of-the-art feature distillation method [49] using diverse lightweight backbones and architecture variations. Interestingly, our approach is orthogonal to feature distillation, and we show that combining it with the state-of-the-art approach of [49] further boosts the performance of student network. To show the generality of our approach beyond keypoint prediction, we then apply it to the SOTA dense prediction-based method, ZebraPose [40], to align the distributions of dense binary code probabilities. Our experiments evidence that this outperforms training a compact ZebraPose in a standard prediction-to-prediction knowledge distillation fashion.

Our main contributions can be summarized as follows. (i) We investigate for the first time knowledge distillation in the context of 6D pose estimation. (ii) We introduce an approach that aligns the teacher and student distributions of local predictions together with their predicted object segmentation scores. (iii) Our method generalizes to both sparse keypoints and dense predictions 6D pose estimation frameworks. (iv) Our approach can be used in conjunction with feature distillation to further boost the student's performance. Our code is available at https://github.com/GUOShuxuan/kd-6d-pose-adlp.

2. Related Work

6D pose estimation. With the great development and success of deep learning in computer vision [12, 13, 26, 29, 31, 36], many works have explored its use for 6D pose estimation. The first attempts [24, 25, 47] aimed to directly regress the 6D pose from the input RGB image. However, the representation gap between the 2D image and 3D rotation and translation made this task difficult, resulting in limited success. Therefore, most methods currently predict quantities that are closer to the input image space. In particular, several techniques jointly segment the object and predict either the 2D image locations of the corners of the 3D object bounding box [19–21] or the 2D displacements from the cells' center of points on the object's surface [33]; Oberweger *et al.* [32] predict 2D keypoints heatmaps to handle occlusion. Instead of exploiting such sparse 2D keypoints, other methods [7, 28, 45] output dense correspondences between the input image and the object 3D model, typically by predicting a 3D coordinate at every input location containing an object of interest. Recently, the state-of-the-art ZebraPose [40] proposed to replace the prediction of 3D coordinates with that of binary codes encoding such coordinates, yet still producing dense predictions. In any event, the original backbones used by all the above-mentioned methods tend to be cumbersome, making them impractical for deployment in resource-constrained environments. However, replacing these backbones with more compact ones yields a significant performance drop. Here, we address this by introducing a knowledge distillation method for 6D pose estimation applicable to any method outputting local predictions, whether sparse or dense.

Knowledge distillation has been proven effective to transfer information from a deep teacher to a shallow student in several tasks. This trend was initiated in the context of image classification, where Hinton *et al.* [17] guide the student's output using the teacher's class probability distributions, and Romero *et al.* [37], Zagoruyko *et al.* [48] and Tian *et al.* [43] encourage the student's intermediate feature representations to mimic the teacher's ones. Recently, many works have investigated knowledge distillation for other visual recognition tasks, evidencing the benefits of extracting task-driven knowledge. For example, in object detection, Zhang *et al.* [49] adapt the feature distillation strategy of [37] to object detectors; Wang *et al.* [46] restrict the teacher-student feature imitation to regions around the positive anchors; Guo *et al.* [10] decouple the intermediate features and the classification predictions of the positive and negative regions; Guo *et al.* [11] distill detection-related

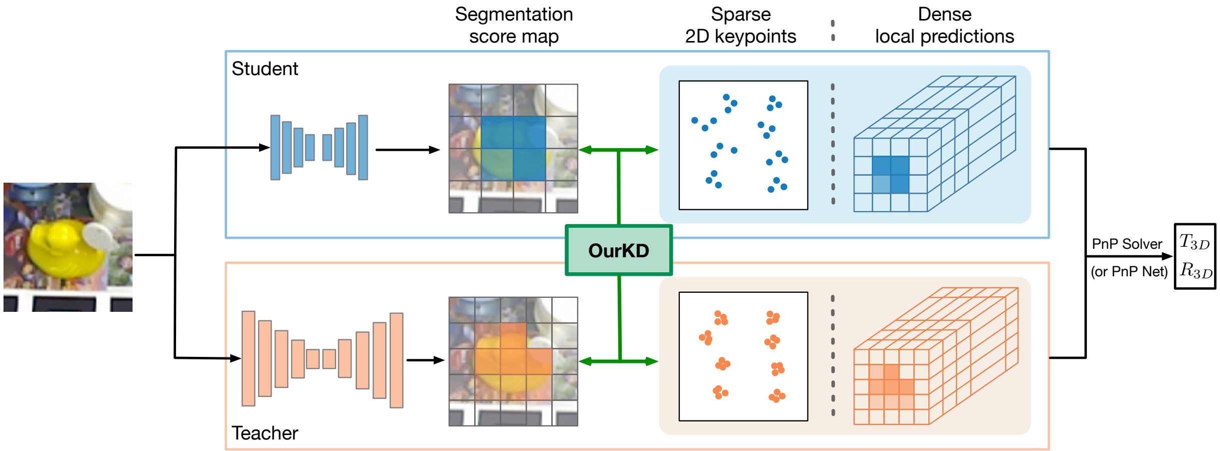

Figure 2. **Overview of our method** (better viewed in color). The teacher and student follow the same general architecture, predicting either sparse 2D keypoints or dense local predictions. Given an RGB input image, they output both a segmentation score map by classifying the individual cells in the feature map, and 2D keypoints voted by each cell as in [21], or one prediction per cell, *e.g*., probabilities of 16D binary codes for ZebraPose [40]. The local predictions, either sparse or dense, then form correspondences, which are passed to a PnP solver [2, 27] or a PnP network [19, 45] to obtain the final 3D translation and 3D rotation. Instead of performing naive prediction-to-prediction distillation, we propose a strategy based on optimal transport that lets us jointly distill the teacher's local prediction distribution with the segmentation score map into the student.

knowledge from a classification teacher to a detection student. In semantic segmentation, Liu *et al*. [30] construct pairwise and holistic segmentation-structured knowledge to transfer. All of these works evidence that task-driven knowledge distillation boosts the performance of compact student models. Here, we do so for the first time for 6D object pose estimation. Note that the concurrent HRPose [9] tackles the scenario where the student and teacher have the same feature dimensions and was evaluated on LINEMOD only. Our work is applicable to more *diverse* student-teacher pairs on more *challenging* benchmarks.

Optimal transport (OT) has received a growing attention both from a theoretical perspective [6, 38, 44] and for specific tasks, including shape matching [41], generative modeling [1], domain adaptation [5], and model fusion [39]. In particular, OT has the advantage of providing a theoretically sound way of comparing multivariate probability distributions without approximating them with parametric models. Furthermore, it can capture more useful information about the nature of the problem by considering the geometric or the distributional properties of the underlying space. Our work constitutes the first attempt at using OT to align the student and teacher local prediction distributions for knowledge distillation in 6D pose estimation.

3. Methodology

Let us now introduce our method to knowledge distillation for 6D pose estimation. As discussed above, we focus on approaches that produce local predictions, such as sparse 2D keypoints [19–21, 33] or dense quantities [7, 28, 40, 45]. In essence, the key to the success of such methods is the prediction of accurate local quantities. However, as shown in Figure 1 for the keypoint case, the predictions of a shallow student network tend to be less precise than those of a deep teacher, i.e., less concentrated around the true keypoint locations in the figure, and thus yield less accurate 6D poses. Below, we first present a naive strategy to distill the teacher's local predictions into the student ones, and then introduce our approach.

3.1. Naive Prediction-to-prediction Distillation

The most straightforward way of performing knowledge distillation is to encourage the student's predictions to match those of the teacher. In our context, one could therefore think of minimizing the distance between the local predictions of the teacher and those of the student. To formalize this, let us assume that the teacher and the student both output N local predictions, i.e., that N cells in the final feature maps participate in the prediction for the object of interest. Then, a naive distillation loss can be expressed as

$$\mathcal{L}_{naive-kd}(P^s, P^t) = \sum_{i=1}^{N} \|P_i^s - P_i^t\|_p \ , \tag{1}$$

where, P_i^s, resp. P_i^t, represent the student's, resp. teacher's, local predictions, and $p \in \{1, 2\}$.

One drawback of this strategy comes from the fact that the teacher and student network may disagree on the number of local predictions they make. For example, as illustrated in Figure 2 for the keypoint case, the number of cells predicted to belong to the object by the student and the teacher may differ. This can be circumvented by only summing over the $\tilde{N} \leq N$ cells that are used by both the teacher and the student. However, the distillation may then be suboptimal, as some student's predictions could potentially be unsupervised by the teacher. Furthermore, and as argued above, a compact student tends to struggle when trained with prediction-to-prediction supervision, and such a naive KD formulation still follows this approach. Therefore, and as will be shown in our experiments, this naive strategy often does not outperform the direct student training, in particular in the sparse 2D keypoints scenario. Below, we therefore introduce a better-suited approach.

3.2. Aligning the Distributions of Local Predictions

In this section, we first discuss our general formulation, and then specialize it to sparse keypoint prediction and dense binary code prediction. As discussed above and illustrated in Figure 2, the number of student local predictions N^s may differ from that of teacher local predictions N^t, preventing a direct match between the individual teacher and student predictions. To address this, and account for the observation that prediction-to-prediction supervision is ill-suited to train the student, we propose to align the *distributions* of the teacher and student local predictions. We achieve this using optimal transport, which lets us handle the case where $N^s \neq N^t$. Formally, to allow the number of student and teacher predictions to differ, we leverage Kantorovich's relaxation [22] of the transportation problem.

Specifically, assuming that all the local predictions have the same probability mass, i.e., $\frac{1}{N^t}$ for the teacher predictions and $\frac{1}{N^s}$ for the student ones, we derive a distillation loss based on Kantorovich's optimal transport problem as

$$\begin{aligned} \bar{\mathcal{L}}_{kd}(P^s, P^t; \pi) &= \min_{\pi} \sum_{i=1}^{N^s} \sum_{j=1}^{N^t} \pi_{ij} \| P_i^s - P_j^t \|_p \\ \text{s.t.} \quad \forall i, \ \sum_{j=1}^{N^t} \pi_{ij} &= \frac{1}{N^s} \ , \quad \forall j, \ \sum_{i=1}^{N^s} \pi_{ij} = \frac{1}{N^t} \ . \end{aligned} \tag{2}$$

In our experiments, we found $p = 2$ to be more effective than $p = 1$ and thus use the ℓ_2 norm below.

The above formulation treats all local predictions equally. However, different predictions coming from different cells in the feature maps might not have the same degree of confidence. In particular, this can be reflected by how confident the network is that a particular cell contains the object of interest, or, in other words, by a segmentation score predicted by the network. Let α_i^s denote such a score for cell i in the student network, and α_j^t a similar score for cell j in the teacher network. We then re-write our distillation loss as

$$\begin{aligned} \tilde{\mathcal{L}}_{kd}(P^s, P^t; \alpha^s, \alpha^t; \pi) &= \min_{\pi} \sum_{i=1}^{N^s} \sum_{j=1}^{N^t} \pi_{ij} \| P_i^s - P_j^t \|_2 \\ \text{s.t.} \quad \forall i, \ \sum_{j=1}^{N^t} \pi_{ij} &= \alpha_i^s \ , \quad \forall j, \ \sum_{i=1}^{N^s} \pi_{ij} = \alpha_j^t \ . \end{aligned} \tag{3}$$

In essence, because this loss involves both the local predictions and the cell-wise segmentation scores, it distills jointly the correspondence-related quantities and the segmentation results from the teacher to the student.

To solve this optimal transport problem, we rely on Sinkhorn's algorithm [6], which introduces a soft versions of the constraints via Kullback-Leibler divergence regularizers. This then yields the final distillation loss

$$\begin{aligned} \mathcal{L}_{kd}(P^s, P^t; \alpha^s, \alpha^t; \pi) = \min_{\pi} & \sum_{i=1}^{N^s} \sum_{j=1}^{N^t} \pi_{ij} \| P_i^s - P_j^t \|_2 \\ & + \varepsilon^2 \mathrm{KL}(\pi, \alpha^s \otimes \alpha^t) + \rho^2 \mathrm{KL}(\pi \mathbf{1}, \alpha^s) \\ & + \rho^2 \mathrm{KL}\left(\pi^\top \mathbf{1}, \alpha^t\right) \ , \end{aligned} \tag{4}$$

where α^s and α^t concatenate the segmentation scores for the student and the teacher, respectively. This formulation was shown to be amenable to fast parallel optimization on GPU platforms, and thus well-suited for deep learning [6,8].

3.2.1 Keypoint Distribution Alignment

Let us now explain how we specialize the formulation in Eq. 4 to the case of a network predicting sparse keypoints. In particular, we consider the case of predicting the 2D locations of the 8 object bounding box corners [19–21, 33]. In this case, we consider separate costs for the 8 individual keypoints, to prevent a 2D location corresponding to one particular corner to be assigned to a different corner.

Let C_k^s and C_k^t denote the predictions made by the student and the teacher, respectively, for the k^{th} 2D keypoint location. Then, we express our keypoint distribution distillation loss as

$$\begin{aligned} & \mathcal{L}_{kd}^{kp}(\{C_k^s\}, \{C_k^t\}; \alpha^s, \alpha^t; \{\pi^k\}) \\ & \quad = \sum_{k=1}^{8} \mathcal{L}_{kd}(C_k^s, C_k^t; \alpha^s, \alpha^t; \pi^k). \end{aligned} \tag{5}$$

In our experiments, we normalize the predicted 2D keypoints by the image size to the $[0, 1]^2$ space, and set ε to 0.001 and ρ to 0.5 to handle outliers.

3.2.2 Dense Binary Code Distribution Alignment

To illustrate the case of dense local predictions, we rely on the ZebraPose [40] formalism, which predicts a 16-dimensional binary code probability vector at each cell of the final feature map. To further encode a notion of location in this dense representation, we concatenate the x- and y-coordinate in the feature map to the predicted vectors. Handling such dense representations, however, comes at a higher computational cost and memory footprint than with the previous sparse keypoints. To tackle this, we therefore average pool them over a small square regions.

Formally, let B^s and B^t represent the average-pooled local augmented binary code probabilities predicted by the student and teacher, respectively. Then, we write our dense prediction distribution distillation loss as

$$\mathcal{L}_{kd}^{bc}(B^s, B^t; \alpha^s, \alpha^t; \pi) = \mathcal{L}_{kd}(B^s, B^t; \alpha^s, \alpha^t; \pi). \quad (6)$$

where α^s and α^t also represent the average-pooled segmentation scores for the student and teacher, respectively. In our experiments, we use a pooling size of 8×8. Furthermore, we set ε to 0.0001 and ρ to 0.1 to handle the outliers over the dense predictions of the binary code probabilities.

3.3. Network Architectures

Our approach can be applied to any network that output local predictions. In our experiments, we use WDRNet [21] for the sparse keypoint case and ZebraPose [40] for the dense prediction one. WDRNet employs a feature pyramid to predict the 2D keypoint locations at multiple stages of its decoder network. These multi-stage predictions are then fused by an ensemble-aware sampling strategy, ultimately still resulting in 8 clusters of 2D locations, i.e., one cluster per 3D bounding box corner. To make the WDRNet baseline consistent with the state-of-the-art methods [7,28,40,45], we incorporate a detection pre-processing step that provides an image patch as input to WDRNet. We refer to this as WDRNet+. We will nonetheless show in our experiments that the success of our distillation strategy does not depend on the use of this detector. ZebraPose constitutes the state-of-the-art 6D pose estimation method. It predicts a binary code at each location in the feature map, and uses these codes to build dense 2D-3D correspondences for estimating 6D pose.

In our experiments, the teacher and student networks follow the same general architecture, only differing in their backbones. Note that different backbones may also yield different number of stages in the WDRNet+ feature pyramid, but our distribution matching approach to knowledge distillation is robust to such differences. To train our WDRNet+ and ZebraPose networks, we rely on the standard losses proposed in [21, 40]. When performing distillation to a student network, we complement these loss terms with our distillation loss of either Eq. 5, for the keypoint case, or Eq. 6 for the dense binary code one. To implement the losses, we rely on the GeomLoss library [8].

4. Experiments

In this section, we first discuss our experimental settings, and then demonstrate the effectiveness and generalization ability of our approach on three widely-adopted datasets, LINEMOD [16], Occluded-LINEMOD [3] and YCB-V [47]. Finally, we analyze different aspects of our method and evaluate it on variations of our architecture.

4.1. Experimental Settings

Datasets. We conduct experiments on the standard LINEMOD [16], Occluded-LINEMOD [3] and YCB-V [47] 6D pose estimation benchmarks. The LINEMOD dataset contains around 16000 RGB images depicting 13 objects, with a single object per image. Following [4], we split the data into a training set containing around 200 images per object and a test set containing around 1000 images per object. The Occluded-LINEMOD dataset was introduced as a more challenging version of LINEMOD, where multiple objects heavily occlude each other in each RGB image. It contains 1214 testing images. For training, following standard practice, we use the real images from LINEMOD together with the synthetic ones provided with the dataset and generated using physically-based rendering [18]. YCB-V [47] is a large dataset containing 21 strongly occluded objects observed in 92 video sequences, with a total of 133,827 frames.

Networks. For WDRNet+, we use DarkNet53 [35] as backbone for the teacher model, as in the original WDRNet [21]. For the compact students, we experiment with different lightweight backbones, including DarkNet-tiny [34] and a further reduced model, DarkNet-tiny-H, containing half of the channels of DarkNet-tiny in each layer. For ZebraPose, we use the pre-trained models of [40] with a ResNet34 [13] backbone as teacher networks and use DarkNet-tiny as backbone for the student networks.

Baselines. We compare our method to the direct training of the student without any distillation (Student), the naive knowledge distillation strategy introduced in Section 3.1 (Naive-KD), and the state-of-the-art feature distillation method (FKD) [49], which, although only demonstrated for object detection, is applicable to 6D pose estimation. For these baselines, we report the results obtained with the best hyper-parameter values. Specifically, for FKD, the best distillation loss weight on all three datasets was 0.01; for Naive-KD, the best weight was 0.1, and the best norm was $p = 1$ for DarkNet-tiny and $p = 2$ for DarkNet-tiny-H, respectively. For our method, the distillation loss was set to 5 for LINEMOD and to 0.1 for both Occluded-LINEMOD and YCB-V. With ZebraPose, we conduct experiments on Occluded-LINEMOD only because of its much larger computational cost, taking many more iterations to converge than WDRNet+ (380K VS 200K). We use a distillation

Table 1. **Results of DarkNet-tiny and DarkNet-tiny-H backbone on LINEMOD dataset with WDRNet+.** We report the ADD-0.1d for the baseline model, Naive-KD, FKD [49] and our KD method for each class. Our method not only outperforms Naive-KD and FKD, but can also be combined with FKD to obtain a further performance boost, yielding state-of-the-art results.

Class	Teacher	DarkNet-tiny					DarkNet-tiny-H				
		Student	Naive-KD	FKD	Ours	Ours+	Student	Naive-KD	FKD	Ours	Ours+
Ape	82.6	73.4	74.1	74.8	74.7	**76.2**	65.4	64.1	68.4	69.4	**69.9**
Bvise	95.5	95.2	95.4	94.2	95.5	**96.7**	92.0	91.4	92.8	**93.8**	93.7
Cam	93.8	91.2	89.7	91.3	91.3	**92.0**	78.4	79.1	83.8	84.5	**84.5**
Can	95.7	92.4	92.7	**94.4**	92.2	94.0	82.2	81.0	83.3	83.9	**83.9**
Cat	92.0	87.2	85.0	87.5	88.4	**88.6**	81.5	78.7	80.7	**81.8**	81.6
Driller	94.8	92.2	93.1	94.8	93.3	**94.8**	85.5	87.4	**90.5**	90.0	90.3
Duck	76.0	70.9	74.4	73.6	73.5	**74.7**	64.3	63.6	66.8	66.5	**68.9**
Eggbox*	99.1	99.3	98.7	98.9	99.1	**99.3**	95.8	95.0	96.3	96.4	**96.4**
Glue*	96.4	97.2	97.1	96.2	97.7	**97.7**	90.7	91.2	91.0	91.9	**93.2**
Holep	86.2	78.0	82.1	79.5	**82.4**	82.2	73.2	72.3	**77.5**	74.1	76.3
Iron	93.6	92.1	92.1	91.4	**93.5**	93.2	86.3	86.3	87.6	88.7	**90.5**
Lamp	97.7	96.6	95.3	96.9	**97.0**	96.8	93.6	94.2	93.4	**94.8**	94.6
Phone	91.2	87.5	88.4	89.4	88.2	**89.6**	76.0	75.8	**80.6**	78.2	79.2
AVG.	91.9	88.7	89.1 (↑ 0.4)	89.4 (↑ 0.7)	89.9 (↑ 1.2)	**90.4** (↑ **1.7**)	81.9	81.6 (↓ 0.3)	84.1 (↑ 2.2)	84.2 (↑ 2.3)	**84.8** (↑ **2.9**)

† Ours+: Ours+FKD distills both the predictions and the intermediate feature maps.

weight of 1.0 for Naive-KD and of 100.0 for our method. We provide the results of the hyper-parameter search in the supplementary material.

Evaluation metric. We report our results using the standard ADD-0.1d metric. It encodes the percentage of images for which the average 3D point-to-point distance between the object model in the ground-truth pose and in the predicted one is less than 10% of the object diameter. For symmetric objects, the point-to-point distances are computed between the nearest points. Note that, on LINEMOD, we report the results obtained using the ground-truth 2D bounding boxes to remove the effects of the pretrained detectors. On Occluded-LINEMOD and YCB-V, we report the results obtained with the same detector as in [7, 40, 45] to evidence the effectiveness of our knowledge distillation method.

4.2. Experiments with WDRNet+

Let us first consider the case of 2D keypoints with WDRNet+. In this scenario, we compare our keypoint distribution alignment method with the Naive-KD and the state-of-the-art feature distillation FKD with multiple student architectures on all three datasets.

Results on LINEMOD. We report the results of our method and the baselines for all classes of the LINEMOD dataset in Table 1 for DarkNet-tiny and DarkNet-tiny-H. While Naive-KD slightly improves direct student training with the DarkNet-tiny backbone, it degrades the performance with DarkNet-tiny-H. This matches our analysis in Section 3; the fact that the student's and teacher's active cells differ make keypoint-to-keypoint distillation ill-suited.

Both FKD and our approach boost the student's results, with a slight advantage for our approach. In particular the accuracy improvement is larger, i.e., 2.3 points, for the smaller DarkNet-tiny-H backbone, for which the gap between the student and the teacher performance is also bigger. Note that the improvement of our approach over the student is consistent across the 13 objects. Interestingly, the types of distillation performed by FKD and by our approach are orthogonal; FKD distills the intermediate features while we distill the predictions. As such, the two methods can be used together. As can be seen from the table, this further boosts the results, reaching a margin over the student of 1.7 points and 2.9 points with DarkNet-tiny and DarkNet-tiny-H, respectively, and thus constituting the state of the art on the LINEMOD dataset for such compact architectures.

Table 2. **Results on OCC-LINEMOD with WDRNet+.** We report the ADD-0.1d for each class. Our method performs on par with FKD [49], combining it with FKD yields a further performance boost.

Class	Teacher	Student	FKD	Ours	Ours+
Ape	33.4	25.5	26.7	25.7	**26.9**
Can	70.9	46.6	53.9	53.5	**54.7**
Cat	45.1	31.4	31.1	32.2	**32.9**
Driller	70.9	51.2	52.1	52.9	**52.9**
Duck	27.0	22.5	25.3	25.7	**27.0**
Eggbox*	53.7	43.4	49.0	48.2	**50.0**
Glue*	70.7	54.5	55.6	55.8	**56.9**
Holep	59.7	49.3	52.2	52.1	**54.5**
AVG.	53.9	40.5	43.2 (↑ 2.7)	43.2 (↑ 2.7)	**44.5** (↑ **4.0**)

Table 3. **Average results on YCB-V with WDRNet+.** Our method outperforms FKD [49] and further boosts the performance combining with it.

Teacher	Student	FKD	Ours	Ours+
46.9	16.1	17.4 (↑ 1.3)	**18.7** (↑ 2.6)	**19.2** (↑ **3.1)**

Table 4. **Results on OCC-LINEMOD with ZebraPose [40]** We report the ADD-0.1d for each class. Our method outperforms Naive-KD and FKD with ZebraPose, showing the generality of our approach to the dense prediction based method.

Class	Teacher	Student	Naive-KD	FKD	Ours
Ape	57.9	47.2	51.1	51.3	**52.0**
Can	95.0	93.2	93.5	**94.5**	94.2
Cat	60.6	53.1	53.9	54.2	**55.2**
Driller	94.8	90.3	90.0	89.9	**90.4**
Duck	64.5	57.2	60.7	60.6	**61.0**
Eggbox*	70.9	69.6	70.0	70.2	**70.7**
Glue*	88.7	84.1	83.7	83.8	**84.3**
Holep	83.0	75.8	78.3	78.2	**78.8**
AVG.	76.9	71.4	72.6 (↑ 1.2)	72.8 (↑ 1.4)	**73.3** (↑ **1.9)**

Results on Occluded-LINEMOD. Let us now evaluate our method on the more challenging Occluded-LINEMOD. Here, we use only FKD [49] as baseline and drop Naive-KD due to its inferior performance shown before. The results are provided in Table 2. Our keypoint-based knowledge distillation method yields results on par with the feature-based FKD on average. Note, however that FKD requires designing additional adaptive layers to match the misaligned feature maps, while our method does not incur additional parameters. More importantly, jointly using our method with FKD achieves the best results with 4.0 points improvements over the baseline student model. For some classes, such as *can*, *eggbox* and *holepuncher*, the boost surpasses 5 points.

Results on YCB-V. The results on the large YCB-V datasets are provided in Table 3. Our method outperforms the baseline and FKD by 2.6 and 1.3 on average. Moreover, the performance is boosted to 19.2 with Ours+FKD. These results further evidence the effectiveness of our method.

4.3. Experiments with ZebraPose

In Table 4, we show the effectiveness of our method when applied to the SOTA dense prediction network ZebraPose [40]. We compare our knowledge distillation strategy with the Naive-KD and FKD. In this dense prediction case, Naive-KD and FKD improve the baselines. Nevertheless, as evidenced by the results, our approaches outperforms both Naive-KD and FKD by 0.7 and 0.5 on average, respectively. This shows the generality of our KD method based on the alignment of local prediction distributions.

Table 5. **Ablation study on LINEMOD: With vs without segmentation scores.**

Model	#Param(M)	ADD-0.1d
WDRNet+(tiny)	8.5	88.7
Ours-NoScores	8.5	89.1
Ours	8.5	**89.9**
WDRNet+(tiny-H)	2.3	81.9
Ours-NoScores	2.3	83.1
Ours	2.3	**84.2**

4.4. Additional Analysis

Let us now further analyze the behavior of our knowledge distillation. The experiments in this section were performed using WDRNet+ on the LINEMOD dataset.

With vs without segmentation scores. We compare the results of our approach without and with the use of the segmentation scores in the optimal transport formulation, i.e., Eq. 2 vs Eq. 3. The comparison in Table 5 shows the benefits of jointly distilling the local predictions and the segmentation scores.

Without detection pre-processing. Note that we incorporated the pre-processing detection step in WDRNet only because it has been shown to boost the pose estimation results. However, the success of our knowledge distillation strategy does not depend on it. To demonstrate this, in the left portion of Table 6, we report the results of our approach applied to the original WDRNet with a DarkNet-tiny backbone. As a matter of fact, the gap between direct student training and our approach is even larger (1.2 vs 2.1), showing the benefits of our approach on weaker networks.

With a simple PnP network. In the right portion of Table 6, we compare the results of our approach with those of the baselines on an architecture obtained by incorporating a simple PnP network at the end of WDRNet, following the strategy of [19]. With such an architecture, the 2D keypoint locations only represent an intermediate output of the network, with the PnP module directly predicting the final 3D translation and 3D rotation from them. As can be seen from these results, our distillation strategy still effectively boosts the performance of the student with this modified architecture, further showing the generality of our approach, which can distill keypoint-based knowledge both for PnP solvers and PnP networks.

Qualitative analysis. We further provide visual comparisons of the predicted 2D keypoints distributions obtained with the baseline student model and with our distilled model on several examples from Occluded-LINEMOD. As shown in Figure 3, the predicted 2D keypoints clusters from our distilled models are closer to the ground-truth object corners than those of the baseline model. Furthermore, our distilled model mimics the teacher's keypoints distributions.

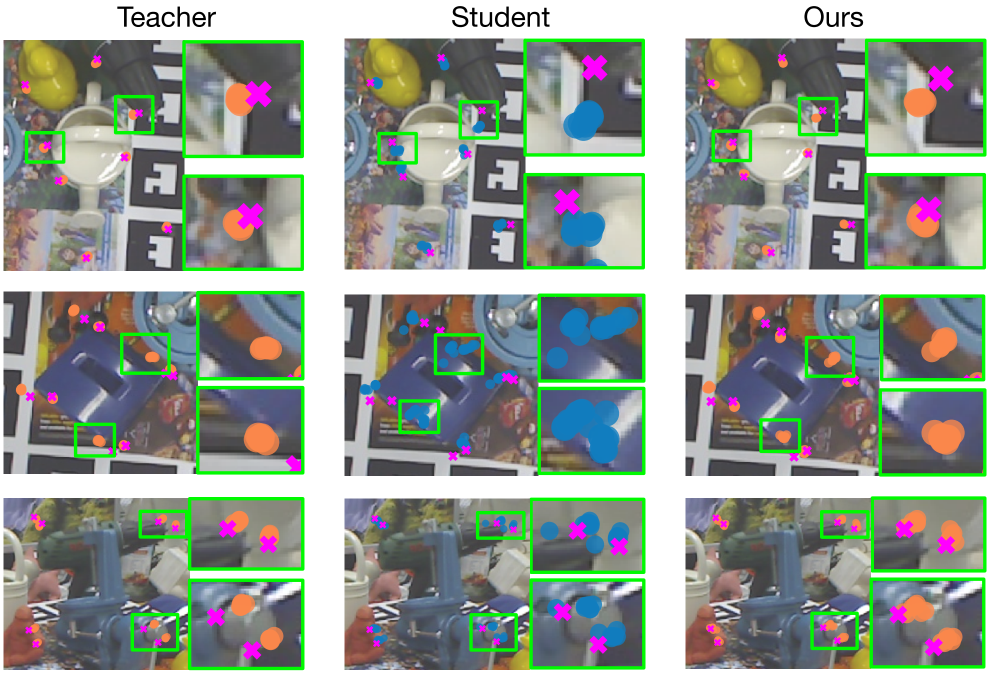

Figure 3. **Qualitative Analysis** (better viewed in color). Comparison of the 2D keypoints predicted with our distilled model (3rd column with orange dots) and the baseline student model (2nd column with blue dots). With our distillation method, the model predicts tighter keypoint clusters, closer to the ground-truth corners (pink crosses) than the baseline model. Furthermore, our distilled model is able to mimic the teacher's keypoint distributions (1st column with orange dots). The light-green boxes highlight some keypoint clusters, which are also zoomed in on the side of the image.

Table 6. **Evaluation under different network settings on LINEMOD.** We report the ADD-0.1d with the original WDRNet framework [21] and with an additional simple PnP network [19]. Our method improves the performance of the student network in both settings.

Class	WDRNet			WDRNet + PnPNet		
	Teacher	Student	Ours	Teacher	Student	Ours
Ape	70.3	41.2	43.0	50.6	29.4	35.1
Bvis	94.2	81.5	86.1	91.7	72.9	80.8
Cam	89.0	67.6	69.8	90.5	56.1	73.3
Can	90.6	72.1	73.8	88.3	57.5	75.9
Cat	87.1	54.3	61.5	62.5	61.8	48.5
Driller	93.6	78.3	79.3	87.1	68.6	71.9
Duck	64.5	35.9	39.6	38.1	32.0	39.6
Eggbox*	95.4	79.3	83.8	99.3	91.8	96.6
Glue*	93.4	83.4	82.7	92.8	87.3	92.2
Holep	77.1	44.2	46.9	70.9	46.4	49.9
Iron	90.9	75.8	75.1	93.3	76.1	80.3
Lamp	96.3	84.8	86.8	95.8	68.7	87.2
Phone	85.3	69.6	67.3	92.3	57.0	76.6
AVG.	86.7	66.8	**68.9** **(↑ 2.1)**	81.0	62.0	**69.8** **(↑ 7.8)**

Limitations. Because of the OT algorithm, training with our method comes with an overhead. Note, however, that we have observed this to have a negligible impact on the actual training clock time. Furthermore, inference comes with no additional cost, and our distilled student model yields better performance. We have also observed that different classes benefit differently from distillation. This raises the possibility of designing class-wise distillation strategy, which we believe could be an interesting direction to explore in the future.

5. Conclusion

We have introduced the first approach to knowledge distillation for 6D pose estimation. Our method is driven by matching the distributions of local predictions from a deep teacher network to a compact student one. We have formulated this as an optimal transport problem that lets us jointly distill the local predictions and the classification scores that segment the object in the image. Our approach is general and can be applied to any 6D pose estimation framework that outputs multiple local predictions. We have illustrated this with the sparse keypoint case and the dense binary code one. Our experiments have demonstrated the effectiveness of our method and its benefits over a naive prediction-to-prediction distillation strategy. Furthermore, our formalism is complementary to feature distillation strategies and can further boost its performance. In essence, our work confirms the importance of developing task-driven knowledge distillation methods, and we hope that it will motivate others to pursue research in this direction, may it be for 6D pose estimation or for other tasks.

Acknowledgement. This work is supported by the Swiss National Science Foundation.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*, 2017. 3

[2] Dániel Baráth and Jiri Matas. Progressive-X: Efficient, Anytime, Multi-Model Fitting Algorithm. In *International Conference on Computer Vision*, 2019. 1, 3

[3] 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. 2, 5

[4] 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. 5

[5] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal Transport for Domain Adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 3

[6] Marco Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems*, 2013. 2, 3, 4

[7] 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, 2, 3, 5, 6

[8] 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. 4, 5

[9] 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

[10] Jianyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu. Distilling Object Detectors via Decoupled Features. *Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2

[11] Shuxuan Guo, Jose M Alvarez, and Mathieu Salzmann. Distilling Image Classifiers in Object Detectors. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *International Conference on Computer Vision*, 2017. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016. 2, 5

[14] Tong He, Chunhua Shen, Zhi Tian, Dong Gong, Changming Sun, and Youliang Yan. Knowledge Adaptation for Efficient Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2019. 1

[15] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A Comprehensive Overhaul of Feature Distillation. In *International Conference on Computer Vision*, 2019. 1

[16] 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. 2, 5

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv Preprint*, 2015. 1, 2

[18] 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. 5

[19] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-Stage 6D Object Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 3, 4, 7, 8

[20] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6D Object Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 3, 4

[21] 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, 2, 3, 4, 5, 8

[22] Leonid Kantorovitch. On the Translocation of Masses. *Management science*, 1958. 4

[23] Tong Ke and Stergios I Roumeliotis. An Efficient Algebraic Solution to the Perspective-Three-Point Problem. In *Conference on Computer Vision and Pattern Recognition*, 2017. 1

[24] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In *International Conference on Computer Vision*, 2017. 2

[25] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *International Conference on Computer Vision*, 2015. 1, 2

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, 2012. 2

[27] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An Accurate O(n) Solution to the PnP Problem. In *International Journal of Computer Vision*, 2009. 1, 3

[28] Zhigang Li, Gu Wang, and Xiangyang Ji. CDPN: Coordinates-based Disentangled Pose Network for Real-time RGB-based 6-DoF Object Pose Estimation. In *International Conference on Computer Vision*, 2019. 1, 2, 3, 5

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single Shot Multibox Detector. In *European Conference on Computer Vision*, 2016. 2

[30] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured Knowledge Distillation for

Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2019. 1, 3

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2015. 2

[32] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making Deep Heatmaps Robust to Partial Occlusions for 3D Object Pose Estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018. 2

[33] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-wise Voting Network for 6DoF Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 3, 4

[34] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Conference on Computer Vision and Pattern Recognition*, 2016. 5

[35] Joseph Redmon and Ali Farhadi. Yolov3: An Incremental Improvement. *arXiv Preprint*, 2018. 5

[36] Eduardo Romera, José M Alvarez, Luis M Bergasa, and Roberto Arroyo. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 2017. 2

[37] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for Thin Deep Nets. *arXiv Preprint*, 2014. 1, 2

[38] Filippo Santambrogio. Optimal Transport for Applied Mathematicians. *Birkäuser, NY*, 2015. 3

[39] Sidak Pal Singh and Martin Jaggi. Model Fusion via Optimal Transport. In *Advances in Neural Information Processing Systems*, 2020. 3

[40] 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, 2, 3, 4, 5, 6, 7

[41] Zhengyu Su, Yalin Wang, Rui Shi, Wei Zeng, Jian Sun, Feng Luo, and Xianfeng Gu. Optimal Mass Transport for Shape Matching and Comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. 3

[42] George Terzakis and Manolis Lourakis. A Consistently Fast and Globally Optimal Solution to the Perspective-n-Point Problem. In *European Conference on Computer Vision*, 2020. 1

[43] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive Representation Distillation. In *International Conference on Learning Representations*, 2020. 2

[44] Cédric Villani. *Optimal Transport: Old and New*, volume 338. Springer, 2009. 2, 3

[45] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. GDR-Net: Geometry-Guided Direct Regression Network for Monocular 6D Object Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 3, 5, 6

[46] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling Object Detectors with Fine-Grained Feature Imitation. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2

[47] 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, 2, 5

[48] Sergey Zagoruyko and Nikos Komodakis. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *International Conference on Learning Representations*, 2017. 1, 2

[49] 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, 5, 6, 7