Knowledge Distillation for 6D Pose Estimation by Keypoint Distribution Alignment

Shuxuan Guo¹ Yinlin Hu¹ Jose M. Alvarez² Mathieu Salzmann¹
¹CVLab, EPFL, Lausanne 1015, Switzerland
²NVIDIA, Santa Clara, CA 95051, USA
{shuxuan.guo, yinlin.hu, mathieu.salzmann}@epfl.ch, josea@nvidia.com

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 for 6D pose estimation. Specifically, we follow a standard approach to 6D pose estimation, consisting of predicting the 2D image locations of object keypoints. In this context, we observe the compact student network to struggle predicting precise 2D keypoint locations. Therefore, to address this, instead of training the student with keypoint-to-keypoint supervision, we introduce a strategy based the optimal transport theory that distills the teacher’s keypoint distribution 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.

1 Introduction

Estimating the 3D position and 3D orientation, a.k.a. 6D pose, of an object relative to the camera from a single image has a longstanding history in computer vision, with many real-world applications, such as robotics, autonomous navigation, and virtual and augmented reality. While modern methods now all rely on deep networks to address this task, the most effective ones 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. In particular, the state-of-the-art methods, GDR-Net [40] and SO-Pose [6], achieve this by combining an object detection module, a large encoder-decoder to predict multiple intermediate dense representations, including 2D-3D correspondences, and a learnable PnP module to output the final pose. Unfortunately, the intermediate dense feature maps they extract incur many parameters, thus yielding huge models that are impractical for deployment on embedded platforms and edge devices. In principle, one could reduce the models’ size by employing smaller backbones. However, as shown in Figure 1 for SO-Pose, the resulting accuracy drops quickly.

By contrast, we have observed keypoint-based methods [17, 30, 18, 19] to yield better robustness to the use of lightweight backbones. This is illustrated in Figure 1 for WDRNet [19] whose performance undergoes a much smaller drop than that of SO-Pose as the number of parameter decreases. We believe this to be due to the fact that these methods aggregate multiple votes, originating from the individual feature-map locations containing the object, for the 2D image positions of a sparse set of 3D keypoints, as shown in Figure 2 where the keypoints are the 8 corners of the 3D object bounding box. Nevertheless, the resulting compact architectures still incur a significant drop in pose accuracy.
In this paper, we address this by introducing a knowledge distillation strategy for such keypoint-based 6D pose estimation networks.

Knowledge distillation aims to transfer some information from a deep teacher network to a compact student one. The research on this topic has tackled diverse tasks, such as image classification [15, 43, 34], object detection [44, 9, 8] and semantic segmentation [27, 12]. While some techniques, such as feature distillation [34, 44, 43, 13], 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 observation. As shown in Figure 2 predicting accurate keypoint locations with keypoint-to-keypoint supervision is much harder for a compact student network than for a deep teacher one. We therefore argue that knowledge distillation for 6D pose estimation should not be performed by matching the individual student and teacher keypoints but instead by encouraging the student’s keypoint distribution to become similar to the teacher one, which leaves more flexibility to the student and thus facilitates its training. To achieve this, we follow an Optimal Transport (OT) formalism [39], which lets us measure the distance between the two keypoint sets. We express this as a loss function that can be minimized using a weight-based variant of Sinkhorn’s algorithm [5], 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 predicted keypoints within a cluster, 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 [14] and Occluded-LINEMOD [2] datasets. Our keypoint distribution alignment strategy consistently outperforms both a keypoint-to-keypoint distillation baseline and the state-of-the-art feature distillation method [44] 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 [44] further boosts the performance of student network.

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

---

1WDRNet+ is a modified version of WDRNet [19] that incorporates a detection step to be consistent with SO-Pose [6].
teacher and student keypoint distributions together with their predicted object segmentation scores.

(iii) Our approach can be used in conjunction with feature distillation to further boost the student’s performance. We will make our code publicly available.

2 Related Work

6D pose estimation. With the great development and success of deep learning in computer vision [23, 10, 26, 11, 33, 28], many works have explored its use for 6D pose estimation. The first attempts [42, 22, 21] 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, the state-of-the-art approaches [25, 20, 6] 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. A consequence of the resulting accuracy boost, however, is a significant increase in the number of learnable parameters, arising from the use of larger encoder-decoder architectures to obtain dense intermediate representations in the two best-performing frameworks, namely GDRNet [40] and SO-Pose [6]. Although we have tried to reduce the size of these models, as shown in Figure 1 for SO-Pose, we have observed a rapid performance drop, even when retaining a relatively large number of parameters. By contrast, we have found that networks predicting a sparser set of 2D-to-3D correspondences, such as WDRNet+ [19] in Figure 1 offered a better robustness to the use of lightweight backbones. In essence, these methods jointly segment the object by classifying learned local features and predict either the image locations [17, 19] or the 2D displacements from the cells’ center [30] or 3D object keypoints, typically taken as the corners of the object bounding box. Nevertheless, the original backbones used by these methods remain cumbersome, and reducing them yields a performance drop. Here, we address this by introducing a knowledge distillation for keypoint-based 6D pose estimation.

Knowledge distillation. 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. [15] guide the student’s output using the teacher’s class probability distributions, and Romero et al. [34], Zagoruyko and Komodakis [43], Tian et al. [38] encourage the student’s intermediate feature representations to mimic the teacher’s ones. Recently, many works have investigated knowledge distillation for other tasks, evidencing the benefits of extracting task-driven knowledge. For example, in object detection, Zhang and Ma [44] adapt the feature distillation strategy of [34] to object detectors; Wang et al. [41] restrict the teacher-student feature imitation to regions around the positive anchors; Guo et al. [8] decouple the intermediate features and the classification predictions of the positive and negative regions; Guo et al. [9] distill detection-related knowledge from a classification teacher to a detection student. In semantic segmentation, Liu et al. [27] 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. Note that the feature distillation strategy of Zhang and Ma [44], although developed for object detection, is general and can be applied to 6D pose estimation. However, such a general distillation method does not leverage the specific properties of 6D pose estimation. Here, we introduce an approach that does so, and show that it outperforms the general FKD method [44] and can be combined with it to obtain a further performance boost.

Optimal transport (OT) has received a growing attention both from a theoretical perspective [39, 5, 35] and for specific tasks, including shape matching [37], generative modeling [1], domain adaptation [4], and model fusion [36]. In particular, OT has the advantage of providing a sound and theoretically grounded way of comparing multivariate probability distributions without the need for approximating them with parametric models. Furthermore, by considering the geometry of the underlying space through a cost metric, it can encode useful information about the nature of the problem. Our work constitutes the first attempt at using OT to align student and teacher keypoint 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 follow a keypoint-based approach [17, 30, 18, 19] to 6D pose estimation. Given a single
Figure 3: **Overview of our method** (better viewed in color). The teacher and student follow a segmentation-driven approach to 6D pose estimation, which, given an RGB input image, outputs both a segmentation score map by classifying the individual cells in the feature map and 2D keypoint locations, with each cell voting for the locations of the 8 corners of the 3D object bounding box. The keypoint locations with the bounding box corners form correspondences, which are passed to a RANSAC-based PnP solver [24] or a simple PnP network [18] to obtain the final 3D translation and 3D rotation. Instead of performing keypoint-to-keypoint distillation, we propose an optimal transport-based strategy that lets us jointly distill the teacher’s keypoint distribution and segmentation score map into the student.

RGB image captured by a calibrated camera, this approach aims to predict the 2D locations of 3D object keypoints in the image plane. This is illustrated in Figure 3 for keypoints taken as the 8 corners of the 3D object bounding box. More precisely, the deep network classifies each local cell, i.e., anchor, in the feature map to obtain a rough object segmentation, and predict the 2D keypoint locations for each active anchor, i.e., each anchor classified as belonging to the object. As such, each bounding box corner corresponds to a cluster of 2D locations in Figure 2. These locations then form 2D-to-3D correspondences that can act as input to a RANSAC-based PnP solver, e.g., [24] or to a shallow PnP network [18] to estimate the 6D pose.

In essence, the key to the success of such a correspondence-based approach is the prediction of accurate 2D keypoint locations. However, as shown in Figure 2, the predictions of a shallow student network differ significantly from those of a deep teacher one; they are less concentrated around the true keypoint locations, and thus yield less accurate 6D poses. Below, we first present a naive distillation strategy to addressing this, and then introduce our approach.

### 3.1 Naive Keypoint-to-keypoint 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 locations predicted by the teacher and those predicted by the student. Formally, with $N$ anchors in the feature map, this can be expressed as

$$
\mathcal{L}_{\text{naive-\text{kd}}}(P_s, P_t) = \sum_{i=1}^{N} \lambda_i \sum_{k=1}^{8} \| P_{ik}^s - P_{ik}^t \|_p ,
$$

(1)

where $P_{ik}^s$, resp. $P_{ik}^t$, represent the student’s, resp. teacher’s, 2D prediction for keypoint $k$ in anchor $i$, $\lambda_i \in \{0, 1\}$ indicates whether the anchor is active or not, and $p \in \{1, 2\}$.

One drawback of this strategy comes from the fact that the teacher and student network may disagree on which anchors are active and which are not, as illustrated in Figure 3. In other words, the $\lambda_i$s may be taken from either the student, the teacher, or the intersection of their segmentation results. However, in any case, the distillation may be suboptimal, as some student’s predictions would be either unsupervised by the teacher or supervised by potentially unconfident teacher predictions.
Furthermore, and as argued above, a compact student tends to struggle when trained with keypoint-to-keypoint supervision, and such a naive KD formulation still follows this approach. Therefore, and as will be shown in our experiments, this naive strategy does not outperform the direct student training. Below, we therefore introduce a better-suited approach to keypoint-based knowledge distillation.

3.2 Keypoint Distribution Alignment

As discussed above and illustrated in Figure 3, the number of active student anchors \(N_s\) may differ from that of active teacher anchors \(N_t\), making a direct match between the individual teacher and student predictions ill-suited. To address this, and account for the observation that keypoint-to-keypoint supervision is ill-suited to train the student, we propose to align the distributions of the teacher and student keypoint predictions. Specifically, we achieve this using optimal transport, which can handle the case where \(N_s \neq N_t\).

Formally, to allow the number of student and teacher anchors to differ, we leverage Kantorovich’s relaxation [20] of the transportation problem. In our context, assuming that all the keypoints have the same probability mass, i.e., \(\frac{1}{N_s}\) for the teacher predictions and \(\frac{1}{N_t}\) for the student ones, we derive a distillation loss based on Kantorovich’s optimal transport problem as

\[
\mathcal{L}_{kd}(P^s, P^t) = \sum_{k=1}^{8} \min_{\pi^s_k} \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} \pi^s_{ik} \| P^s_{ik} - P^t_{jk} \|_2
\]

s.t. \(\forall k, i, \sum_{j=1}^{N_t} \pi^t_{ij} = \frac{1}{N_s}, \forall k, j, \sum_{i=1}^{N_s} \pi^s_{ij} = \frac{1}{N_t}\).

Note that we consider separate costs for the individual keypoints, thus preventing a 2D location corresponding to one particular corner to be assigned to a different corner. In our experiments, we found \(p = 2\) to be more effective than \(p = 1\) and thus the \(\ell_2\) norm below.

The above formulation treats all keypoint predictions equally. However, different predictions coming from different anchors might not have the same degree of confidence. In particular, this can be reflected by how confident the network is that a particular anchor contains the object of interest, or, in other words, by the segmentation score predicted by the network. Let \(\alpha^s_k\) denote such a score for anchor \(i\) in the student network, and \(\alpha^t_j\) a similar score for anchor \(j\) in the teacher network. Then we re-write our distillation loss as

\[
\mathcal{L}_{kd}(P^s, P^t) = \sum_{k=1}^{8} \min_{\pi^s_k} \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} \pi^s_{ik} \alpha^s_i \| P^s_{ik} - P^t_{jk} \|_2
\]

s.t. \(\forall k, i, \sum_{j=1}^{N_t} \pi^t_{ij} = \alpha^s_i, \forall k, j, \sum_{i=1}^{N_s} \pi^s_{ij} = \alpha^t_j\).

In essence, because this loss involves both the 2D keypoint locations and the anchor-wise classification scores, it distills jointly the correspondences and the segmentation results from the teacher to the student.

To solve this optimal transport problem, we rely on Sinkhorn’s algorithm [5], which introduces a soft versions of the contraints via Kullback-Leibler divergence regularizers. This then yields the final distillation loss

\[
\mathcal{L}_{kd}(P^s, P^t) = \sum_{k=1}^{8} \min_{\pi^s_k} \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} \pi^s_{ik} \| P^s_{ik} - P^t_{jk} \|_2
\]

\[
+ \sum_{k=1}^{8} (\varepsilon^2 \text{KL}(\pi^k, \alpha^s \otimes \alpha^t) + \rho^2 \text{KL}(\pi^k, \mathbf{1}, \alpha^s) + \rho^2 \text{KL}(\pi^k, \mathbf{1}, \alpha^t))
\]

where \(\alpha^s\) and \(\alpha^t\) concatenate the classification score values 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 [5][7]. In our experiments, we normalize the predicted 2D keypoints by the image size to the \([0, 1]^2\) space and set \(\varepsilon\) to 0.001, and \(\rho\) to 0.5 to handle outliers.

\[\text{Note that these scores do not depend on the keypoint index } k \text{ as a single score is predicted for each anchor.}\]
3.3 Network Architecture

Our approach can be applied to any network that predicts the 2D locations of 3D object keypoints. Without loss of generality, we use WDRNet [19] as our base network, which is a typical keypoint-based framework. WDRNet employs a feature pyramid strategy 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 our baseline consistent with the state-of-the-art methods [25, 40, 6], we also use a detection pre-processing step that provides an image patch as input to WDRNet. We refer to this as WDRNet+. As will be shown in our results, the pre-processing detection step allows WDRNet to outperform the state-of-the-art SO-Pose [6] with lightweight backbones. Nevertheless, as also evidenced by our experiments, the success of our distillation strategy does not depend on it.

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 feature pyramid, but our distribution matching approach to knowledge distillation is robust to such differences.

To train our WDRNet+ networks, we rely on the focal loss for the segmentation branch and the 3D regression loss proposed in [19] for the keypoint prediction one. When performing distillation to a student network, we complement these loss terms with our distillation loss of Eq. 4. To implement it, we rely on the GeomLoss library [7].

4 Experiments

In this section, we first discuss our experimental settings, and then demonstrate the effectiveness and generalization ability of our approach on two widely-adopted datasets, LINEMOD and Occluded-LINEMOD. 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 [14] and Occluded-LINEMOD [2] 6D pose estimation benchmarks. The LINEMOD dataset contains around 16000 RGB images depicting 13 objects, with a single object per image. Following [3], 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 [16].

Teacher & student architectures. For our teacher models, we use DarkNet53 [31] as backbone, as in the original WDRNet [19]. For the compact students, we experiment with different lightweight backbones, including DarkNet-tiny [32] and a further reduced model, DarkNet-tiny-H, containing half of the channels of DarkNet-tiny in each layer.

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) [44], 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 both 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 Occluded-LINEMOD. 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 is computed as 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
Table 1: Results of DarkNet-tiny backbone on LINEMOD dataset. We report the ADD-0.1d for the baseline model, Naive-KD, FKD [44] 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 | Student | Naive-KD | FKD [44] | Ours | Ours+FKD [44] |
|-------|---------|---------|----------|----------|------|---------------|
| Ape   | 82.6    | 73.4    | 74.1     | 74.8     | 74.7 | 76.2          |
| Bvise | 95.5    | 95.2    | 95.4     | 94.2     | 95.5 | 96.7          |
| Cam   | 93.8    | 91.2    | 89.7     | 91.3     | 91.3 | 92.0          |
| Can   | 95.7    | 92.4    | 92.7     | **94.4** | 92.2 | 94.0          |
| Cat   | 92.0    | 87.2    | 85.0     | 87.5     | 88.4 | **88.6**      |
| Driller | 94.8    | 92.2    | 93.1     | 94.8     | 93.3 | **94.8**      |
| Duck  | 76.0    | 70.9    | 74.4     | 73.6     | 73.5 | **74.7**      |
| Eggbox* | 99.1    | 99.3    | 98.7     | 98.9     | 99.1 | **99.3**      |
| Glue* | 96.4    | 97.2    | 97.1     | 96.2     | 97.7 | **97.7**      |
| Holep | 86.2    | 78.0    | 82.1     | 79.5     | **82.4** | 82.2 |
| Iron  | 93.6    | 92.1    | 92.1     | 91.4     | **93.5** | 93.2 |
| Lamp  | 97.7    | 96.6    | 95.3     | 96.9     | **97.0** | 96.8 |
| Phone | 91.2    | 87.5    | 88.4     | 89.4     | 88.2 | **89.6**      |
| AVG   | 91.9    | 88.7    | 89.1     | 89.4     | 89.9 | 90.4 (↑ 1.7) |

Table 2: Results of DarkNet-tiny-H backbone on LINEMOD dataset. We report the ADD-0.1d for the baseline model, Naive-KD, FKD [44] and our KD method for each class. As in Table 1 we achieve the state-of-the-art results on the reduced tiny-H backbone.

| Class | Teacher | Student | Naive-KD | FKD [44] | Ours | Ours+FKD [44] |
|-------|---------|---------|----------|----------|------|---------------|
| Ape   | 82.6    | 65.4    | 64.1     | 68.4     | 69.4 | **69.9**      |
| Bvise | 95.5    | 92.0    | 91.4     | 92.8     | **93.8** | 93.7 |
| Cam   | 93.8    | 78.4    | 79.1     | 83.8     | 84.5 | **84.5**      |
| Can   | 95.7    | 82.2    | 81.0     | 83.3     | 83.9 | **83.9**      |
| Cat   | 92.0    | 81.5    | 78.7     | 80.7     | **81.8** | 81.6 |
| Driller | 94.8    | 85.5    | 87.4     | **90.5** | 90.0 | 90.3          |
| Duck  | 76.0    | 64.3    | 63.6     | 66.8     | 66.5 | **68.9**      |
| Eggbox* | 99.1    | 95.8    | 95.0     | 96.3     | 96.4 | **96.4**      |
| Glue* | 96.4    | 90.7    | 91.2     | 91.0     | 91.9 | **93.2**      |
| Holep | 86.2    | 73.2    | 72.3     | **77.5** | 74.1 | 76.3          |
| Iron  | 93.6    | 86.3    | 86.3     | 87.6     | 88.7 | **90.5**      |
| Lamp  | 97.7    | 93.6    | 94.2     | 93.4     | **94.8** | 94.6 |
| Phone | 91.2    | 76.0    | 75.8     | **80.6** | 78.2 | 79.2          |
| AVG   | 91.9    | 81.9    | 81.6     | 84.1     | 84.2 | **84.8** (↑ 2.9) |

the effects of the pretrained detectors. On Occluded-LINEMOD, we report the results obtained with the same detector as in [40, 6] to evidence the effectiveness of our knowledge distillation method.

4.2 Comparison with the State of the Art

Results on LINEMOD. We report the results of our method and the baselines for all classes of the LINEMOD dataset in Table 1 and Table 2 for DarkNet-tiny and DarkNet-tiny-H, respectively. 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 anchors 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,
Table 3: **Results on OCC-LINEMOD.** We report the ADD-0.1d for each class. Our method performs on par with FKD, combining it with FKD yields a further performance boost.

| Class    | SO-Pose | Teacher | Student | FKD [44] | Ours | Ours + FKD [44] |
|----------|---------|---------|---------|----------|------|-----------------|
| Ape      | 29.0    | 33.4    | 25.5    | 26.7     | 25.7 | 26.9            |
| Can      | 51.8    | 70.9    | 46.6    | 53.9     | 53.5 | 54.7            |
| Cat      | 19.1    | 45.1    | 31.4    | 31.1     | 32.2 | 32.9            |
| Driller  | 46.2    | 70.9    | 51.2    | 52.1     | 52.9 | 52.9            |
| Duck     | 29.7    | 27.0    | 22.5    | 25.3     | 25.7 | 27.0            |
| Eggbox*  | 31.3    | 53.7    | 43.4    | 49.0     | 48.2 | 50.0            |
| Glue*    | 46.8    | 70.7    | 54.5    | 55.6     | 55.8 | 56.9            |
| Holep    | 45.4    | 59.7    | 49.3    | 52.2     | 52.1 | 54.5            |
| AVG      | 37.3    | 53.9    | 40.5    | 43.2     | 43.2 | 44.5 (↑ 4.0)    |

the types of distillation performed by FKD and by our approach are orthogonal; FKD distills the 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 constitutes the state of the art on the LINEMOD dataset for such compact architectures.

Results on Occluded-LINEMOD. We now show the generality of our approach by evaluating it on the more challenging Occluded-LINEMOD. Here, we use only FKD [44] as baseline and drop Naive-KD due to its inferior performance shown before. The results are provided in Table 3. 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 state-of-the-art results with 4.0 points improvements over the baseline student model. For some classes, such as can, eggbox and holepuncher, the improvements surpasses 5 points.

### 4.3 Additional Analyses

**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. 3 vs Eq. 2. The comparison in Table 4 shows the benefits of jointly distilling the predicted 2D locations and the segmentation scores. In the table, we also report the results of the baseline SO-Pose [6] and WDRNet+. The numbers show that, despite its larger number of weights, SO-Pose yields worse results than our student baseline for a comparable backbone. Moreover, its performance drops significantly from 85.4 to 66.7 as the number of parameters decreases, while that of WDRNet+ degrades more gracefully.

**Without detection pre-processing.** Note that we incorporated the pre-processing detection step in WDRNet only to show that keypoint-based methods could match the performance of the state-of-the-art dense prediction ones, such as SO-Pose, which also estimate the pose based on detected image patches. In other words, the success of our knowledge distillation strategy does not depend on the detection. To demonstrate this, in the left portion of Table 5, 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.

Table 4: **Ablation study on LINEMOD: With vs without segmentation scores.**

| Model     | #Param(M) | ADD-0.1d |
|-----------|-----------|----------|
| SO-Pose†  | 14.2      | 85.4     |
| WDRNet+(tiny) |          |          |
| Ours-NoScores | 8.5      | 88.7     |
| Ours      |           | 89.1     |
| SO-Pose†  | 13.1      | 66.7     |
| WDRNet+(tiny-H) |         |          |
| Ours-NoScores | 2.3      | 81.9     |
| Ours      |           | 83.1     |

† SO-Pose model with the corresponding backbone.
Table 5: **Evaluation under different network settings on LINEMOD.** We report the ADD-0.1d with the original WDRNet framework and with an additional simple PnP network. Our method improves the performance of the student network in both settings.

| Class    | WDRNet [19] | WDRNet + PnPNet [18] |
|----------|-------------|-----------------------|
|          | 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) |

With a simple PnP network. In the right portion of Table 5, 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 [18]. 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.

### 5 Conclusion

We have introduced the first approach to knowledge distillation for 6D pose estimation. Our method is driven by matching the distributions of predicted 2D keypoint locations 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 predicted 2D locations and classification scores that segment the object in the image. Our experiments have demonstrated the effectiveness of our approach and its benefits over a naive point-to-point 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.

**Limitations and societal impact.** Our results show that some object classes benefit more from our KD approach than others. In the future, we will aim to analyze this in more detail, with a view to designing a class-adaptive version of our method. Furthermore, our current framework is designed for keypoint-based 6D pose estimation methods. Although these methods are successful, we intend to study if our distribution-based approach can be extended to frameworks that perform dense prediction for 6D pose estimation.

By achieving state-of-the-art 6D pose estimation results with compact networks, our approach opens the door to deployment on mobile applications, including safety-critical ones, such as pose estimation for autonomous navigation or space debris capture. Nevertheless, our approach does not offer robustness guarantees, in particular to adversarial attacks. As such, our approach could potentially be diverted for harmful objectives.
References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, 2017.

[2] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *European Conference on Computer Vision*, 2014.

[3] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[4] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[5] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, 2013.

[6] Y. Di, F. Manhardt, G. Wang, X. Ji, N. Navab, and F. Tombari. SO-Pose: Exploiting self-occlusion for direct 6d pose estimation. In *International Conference on Computer Vision*, 2021.

[7] J. Feydy, T. Séjourné, F.-X. Vialard, S.-i. Amari, A. Trouve, and G. Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

[8] J. Guo, K. Han, Y. Wang, H. Wu, X. Chen, C. Xu, and C. Xu. Distilling object detectors via decoupled features. *arXiv Preprint*, 2021.

[9] S. Guo, J. M. Alvarez, and M. Salzmann. Distilling image classifiers in object detectors. *Advances in Neural Information Processing Systems*, 34, 2021.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *International Conference on Computer Vision*, 2017.

[12] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan. Knowledge adaptation for efficient semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[13] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi. A comprehensive overhaul of feature distillation. In *International Conference on Computer Vision*, 2019.

[14] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian Conference on Computer Vision*, 2012.

[15] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv Preprint*, 2015.

[16] T. Hodań, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas. BOP challenge 2020 on 6D object localization. *European Conference on Computer Vision Workshops*, 2020.

[17] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann. Segmentation-driven 6d object pose estimation. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[18] Y. Hu, P. Fua, W. Wang, and M. Salzmann. Single-stage 6d object pose estimation. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[19] Y. Hu, S. Speierer, W. Jakob, P. Fua, and M. Salzmann. Wide-depth-range 6d object pose estimation in space. In *Conference on Computer Vision and Pattern Recognition*, 2021.
[20] L. Kantorovitch. On the translocation of masses. *Management science*, 1958.

[21] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. SSD-6D: Making rgb-based 3d detection and 6d pose estimation great again. In *International Conference on Computer Vision*, 2017.

[22] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *International Conference on Computer Vision*, 2015.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[24] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. 2009.

[25] Z. Li, G. Wang, and X. Ji. CDPN: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *International Conference on Computer Vision*, 2019.

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision*, 2016.

[27] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang. Structured knowledge distillation for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. 2017.

[30] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao. Pvenet: Pixel-wise voting network for 6dof pose estimation. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[31] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv Preprint*, 2018.

[32] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[33] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[34] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. FitNets: Hints for thin deep nets. *arXiv Preprint*, 2014.

[35] F. Santambrogio. Optimal transport for applied mathematicians. *Birkhäuser, NY*, 2015.

[36] S. P. Singh and M. Jaggi. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems*, 2020.

[37] Z. Su, Y. Wang, R. Shi, W. Zeng, J. Sun, F. Luo, and X. Gu. Optimal mass transport for shape matching and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[38] Y. Tian, D. Krishnan, and P. Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020.

[39] C. Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

[40] G. Wang, F. Manhardt, F. Tombari, and X. Ji. GDR-Net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[41] T. Wang, L. Yuan, X. Zhang, and J. Feng. Distilling object detectors with fine-grained feature imitation. In *Conference on Computer Vision and Pattern Recognition*, 2019.
A Appendix

A.1 Dataset and Codebase

LINEMOD [14] and Occluded-LINEMOD [2] 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 [3], 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 [16]. In our work, we respect the terms and conditions of use listed on the websites.

WDRNet [19] is an open-source 6D pose estimation codebase built on Pytorch [29], and is released under the non-commercial use license. Together with WDRNet, we also exploit the detector pre-processing portion of the SO-Pose [6] codebase, 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 [7], 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.

A.2 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 [44]. 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.** As shown in Table A1, 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 A2. 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 gets the best performance for DarkNet-tiny. Therefore, we report the best results for each backbone in the main paper.

**FKD [44].** We follow the same strategy as above, and report the results for Ape and Duck with FKD in Table A3. 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 A4, a distillation
Table A1: Results of Naive-KD with DarkNet-tiny-H backbone on Ape and Duck. 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     |
| Ape   | 82.6    | 65.4    | 63.2    | 64.4    |
| Duck  | 76.0    | 64.3    | 59.4    | 63.3    |
| AVG.  | 79.3    | 64.8    | 61.3    | 63.9    |

Table A2: Results of Naive-KD on LINEMOD dataset. We report the ADD-0.1d for the Naive-KD with DarkNet-tiny-H and DarkNet-tiny backbones with $ps$ and weights searched from Table A1.

| Class | Teacher | DarkNet-tiny-H | DarkNet-tiny |
|-------|---------|----------------|--------------|
|       |         | $p = 1$ | $p = 2$ | $p = 1$ | $p = 2$ |
|       |         | 0.1    | 0.1    | 0.1    | 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   |

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.

A.3 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 $p$s and the weight of our distillation loss. As for the baseline, we focused this search on DarkNet-tiny-H for the Ape and Duck classes. As shown in Table A5, $p = 2$ 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 A6, 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.

A.4 Qualitative Analysis

In this section, we provide additional 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 A1, the predicted 2D keypoints clusters from our distilled models are closer to the ground-truth object corners than those of the baseline model. Furthermore,
Table A3: **Weight searching for FKD on LINEMOD dataset (Ape and Duck).** We report the ADD-0.1d for FKD [44] 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 A4: **Results of FKD on Occluded-LINEMOD dataset.** We report the ADD-0.1d for FKD [44] 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 | -   |

Our distilled model mimics the teacher’s keypoints distributions. As a result, our distilled model yields more accurate pose estimates than the baseline student one.
Figure A1: **Qualitative Analysis** (better viewed in color). In (a), we compare the 2D keypoints predictions from our distilled model (3rd column with orange dots) and the baseline student model (2nd column with blue dots). With our proposed keypoint distribution alignment distillation method, the model predicts tighter keypoint clusters, closer to the ground-truth corners (pink cross) than the baseline model. Furthermore, our distilled model is able to mimic the teacher’s keypoint distributions (1st column with orange dots). Light-green boxes highlight some keypoints clusters, which are also zoomed in on the side of the image. In (b), we show the estimated poses. The blue boxes are the predicted 3D bounding boxes while the gray ones are the ground-truth bounding boxes. Our distilled model generates better pose estimates than the student model.
Table A5: **Results of our proposed KD with DarkNet-tiny-H backbone on LINEMOD dataset (Ape and Duck).** We report the ADD-0.1d for our proposed KD with different \( p \)s and weights.

| Class  | Teacher | Student | \( p = 1 \) | \( p = 2 \) |
|--------|---------|---------|-------------|-------------|
|        |         |         | 0.0          | 0.0          |
| Ape    | 82.6    | 65.4    | 61.9 61.5    | 66.5 69.4    |
| Duck   | 76.0    | 64.3    | 61.2 61.9    | 65.1 66.5    |
| AVG.   | 79.3    | 64.8    | 61.6 61.7    | 65.8 67.9    |

Table A6: **Results of our proposed KD on Occluded-LINEMOD dataset.** 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|