Tackling Two Challenges of 6D Object Pose Estimation: Lack of Real Annotated RGB Images and Scalability to Number of Objects

Juil Sock[⋆], Pedro Castro[⋆], Anil Armagan, Guillermo Garcia-Hernando, and Tae-Kyun Kim

Imperial College London, UK

Abstract. State-of-the-art methods for 6D object pose estimation typically train a Deep Neural Network per object, and its training data first comes from a 3D object mesh. Models trained with synthetic data alone do not generalise well, and training a model for multiple objects sharply drops its accuracy. In this work, we address these two main challenges for 6D object pose estimation and investigate viable methods in experiments. For lack of real RGB data with pose annotations, we propose a novel self-supervision method via pose consistency. For scalability to multiple objects, we apply additional parameterisation to a backbone network and distill knowledge from teachers to a student network for model compression. We further evaluate the combination of the two methods for settings where we are given only synthetic data and a single network for multiple objects. In experiments using LINEMOD, LINEMOD OCCLUSION and T-LESS datasets, the methods significantly boost baseline accuracies and are comparable with the upper bounds, i.e., object specific networks trained on real data with pose labels.

1 Introduction

6D object pose estimation is an important research area in the intersection of computer vision and robotics. Thanks to the development of depth cameras and improved learning/non-learning methods, accurate estimation of 6D object poses in unconstrained environments has been achieved. We now investigate back to RGB image input for its ubiquity, and pursue high-precision pose estimation for robotics in real cluttered environments. An obvious recent trend is to use deep neural networks (DNN), which have shown strong performance using only RGB images.

DNN-based methods are trained primarily using rendered images of 3D object models, where pose labels are automatically generated. Since such methods do not generalise well to real testing data, DNNs also exploit real RGB images with accurate 6D pose labels in addition to the synthetic data. The issue here is in a lack of real annotated data. Note that annotating 6D poses in real RGB images in quality and quantity is a hard problem itself. To overcome this limitation, supervised and unsupervised domain adaptation methods (for real—synthetic)

[⋆] equal contribution

have been proposed [3,8,45]. In this work, we propose a novel self-supervised learning method which results in a domain-invariant 6D pose estimator without any form of pose labels of real data.

State-of-the-art methods typically train a neural network per object, ending up with as many such networks as the number of objects. Its scalability is a major issue, and has been overlooked in favor of accuracy. Although for other vision problems such as image classification or object detection, a deep network is able to learn and perform well over numerous object categories, the same thing has not been observed in the literature of 6D object pose estimation, but rather the opposite [24]. In this paper, we examine the issue, and confirm that 6D pose accuracy rapidly drops when increasing the number of objects for a single DNN. We then investigate methods to relieve the problem and present scalable multi-object architectural modifications.

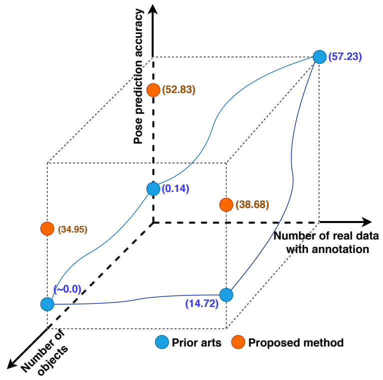


Fig. 1: **Pose estimation accuracy on the two axes by a single network.** While the existing pose estimator largely drops accuracy with less real annotated data and multiple objects, the proposed method retains 60-90% of the best accuracy. (#) indicate the recall rates by ADD on LINEMOD. Best viewed in colour.

Each of the two challenges is tackled in depth and methods are investigated in the experiments using the LINEMOD[15], LINEMOD OCCLUSION[15], and T-LESS[17] datasets. Our methods dramatically improve baseline accuracy and are almost as good as the upper bounds (see Fig. 1). While the blue points are the performance of prior-arts, which sharply drops accuracy for less real training data and more objects. The red points are of our methods, which are far above the respective baselines and almost as good as the upper bound i.e. multiple networks trained by full amount of real data with pose labels. The two proposed approaches are also combined, yielding high accuracy when only synthetic data are used for training and a single DNN is trained for multiple objects.

To our knowledge, our work is first to investigate both challenges together in this level. Besides, the proposed framework can be applied to any state-of-the-art object pose method and barely compromise accuracy while keeping runtime and memory low. *The project codes and data will be made publicly available.*

2 Related Work

Datasets. Annotating geometric labels such as 6D pose requires much effort often involving special hardware [67,23] and only captures limited views as the space of possible view angles is immense. Efforts have been made to create a real RGB dataset with high-quality labels such as Pix3D [56], yet it is limited due to the time and effort required.

Synthetic-real gap. Existing methods render synthetic images from a 3D model with perfect labels to train the network in the synthetic domain and deploy them in real-world [54,26]. However, training purely by synthetic images does not generalise well to real images. Although visually similar, the rendered and real images are different in many ways due to factors such as compression effects or lens vignetting. In some cases, the model used for rendering is obtained by reconstruction of the object from multi-views, which inevitably has considerable geometric error.

Domain adaptation. To overcome this shortcoming, mainly two different approaches have been proposed for 6D object pose estimation: domain randomization [69,57] and domain adaptation [3]. Domain randomization hypothesises that enough variability in augmentation for simulated images will generalise to real images [59]. However, the domain randomisation is limited to the type of parameters (e.g. brightness, contrast, etc.) being randomised, selected heuristically. The chosen parameters might not be relevant to the domain gap. Recently, Zakharov *et al.* [69] proposed a method that learns the optimal weights for a set of type of parameters which helps achieve maximum domain confusion. Domain adaptation is categorised into unsupervised [3,8] or supervised [45] learning. Supervised domain adaptation uses image-label pairs from both the source and target domain and directly learns the mapping between the two representations [9,45]. These methods show promising results but require labels in the target domain. Recently, unsupervised domain adaptation with Generative Adversarial Networks (GANs) [19,3,37] has been proposed to generate target domain images without labels. Another example of an unsupervised method is GRL [8], where domain-invariant features are generated to deceive domain classifiers. However, often the performance of these methods is suboptimal as the methods learn to match the distributions of the domains without considering the task at hand [30]. Also, such methods are prone to overfitting and degeneration of performance is observed if the samples are out of distribution [69,57].

Self-supervision. Recently, self-supervised methods have shown that a model can be trained to predict hand or object pose by constraining the prediction to be consistent with input data by fitting [36,39,63,29]. Although they do not require pose labels, they use an alternative form of ground truth as a weak supervisory signal. [36,39,29] require the ground truth silhouette of objects and [63] exploits depth maps and multi-view data to achieve high performance. Self-supervised learning has also been used to learn more meaningful feature representations by applying simple geometric augmentation to input images such as rotation [60,6]. In this work, we do not use any form of pose ground truth. Instead, we propose to apply the silhouette of predicted pose to augment the input image. This technique is shown to be crucial for bridging the source—domain gap in our experiments.

Multiple objects. Previous work focus on designing a model capable of accurately estimating 6D pose of a single object. Thus, they end up with many models, even when tackling multi-object datasets [42,40,38]. Deploying multiple high capacity models each optimised for a single object, however, is clearly infeasible for mobile/wearable devices. Others claim that simple multi-output prediction is enough to extend the approach to multi-objects [42,66,64,70], which we found to be insufficient, and we investigate methods to scale up. Several existing methods are built on top of a detection step, disentangling the 2D image location of the object from its 3D orientation [57,40,32,38,44], thereby relaxing both tasks. They assume prior object class information is available from the object detector, which is used to select the respective pose estimator, when they have single object architectural designs i.e. a pose estimator per object. Even in multi-object model formulations, the object class prior is only used at inference [40,32,38,44,42].

Additional parameterisation. The most straightforward approach to support multi-objects without significantly increasing the capacity of a model is through careful injection of object specific parameterisation. In multi-task scenarios, the simplest form of additional parameterisation is the addition of task specific parameters at output level [14,49,42]. In the transfer learning domain, efforts have been made in the pursuit of representation learning across multiple tasks or domains [46,2]. A universal parametric network was proposed by [47], where a subset of parameters in intermediate layers is task specific and easily interchangeable. However, their approach introduces a sizeable overhead, doubling the network's original capacity when tackling 10 different tasks. When task selection is available prior to inference, one can design their system to be preemptively conditioned thereby reducing the overhead. Conditional and Adaptive Instance Normalization [7,20] were introduced in style transfer as conditioning strategies for generating images with similar styles by learning style specific normalization parameters, which can be arbitrary changed with a parameterisation lookup table. These parameters add considerably less overhead than prior attempts, both in terms of model capacity and necessary computation, and have been successfully used outside the style transfer domain [68,55].

Model compression. Making use of existing fully trained models, Knowledge Distillation (KD) is one of the primary tools for model compression [16]. In this framework, a small student network can learn through the distilled knowledge of a larger teacher, which in practice are the softened label outputs or the feature distribution of the teacher, thereby transferring the knowledge between networks. Works using multiple teachers, each specialised in a certain domain, have also been successfully proposed [16,33]. KD methods were primarily designed for classification [16,1,50,43] and adapting these methods to regression problems is not trivial [52,5]. Other available compression methods include quantization and pruning. Quantization removes numerical precision through the bit representation reduction of floating point operations, therefore, achieving high throughput with low memory usage and compromising accuracy [11,62,22,43]. Pruning aims at discarding redundant and unimportant network connections, simultaneously reducing unnecessary computation and memory usage [34,13,12] at the cost of pipeline complexity and specialised equipement [65].

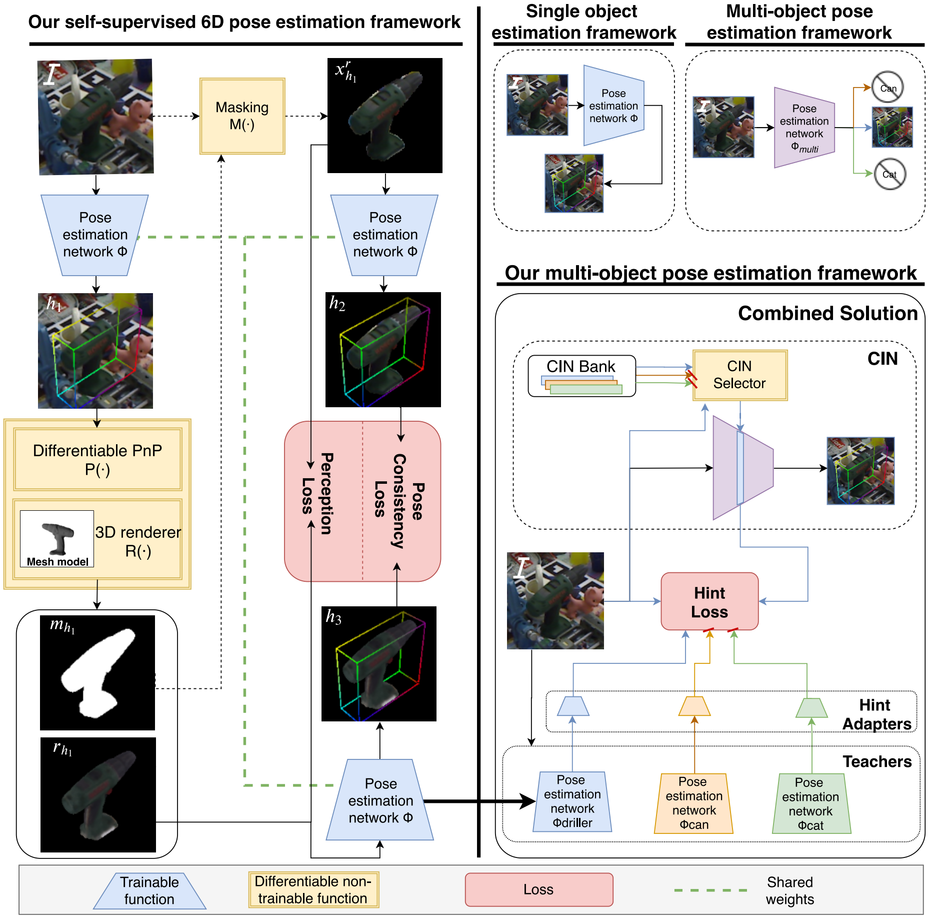

Our self-supervised 6D pose estimation framework
I
Masking M(·)
$x^r_{h_1}$
Pose estimation network Φ
Pose estimation network Φ
h_1
h_2
Differentiable PnP P(·)
Mesh model
3D renderer R(·)
Perception Loss
Pose Consistency Loss
m_{h_1}
h_3
r_{h_1}
Pose estimation network Φ
Single object estimation framework
Pose estimation network Φ
Multi-object pose estimation framework
Pose estimation network Φ_{multi}
Can
Cat
Our multi-object pose estimation framework
Combined Solution
CIN Bank
CIN Selector
CIN
Hint Loss
Hint Adapters
Teachers
Pose estimation network Φdriller
Pose estimation network Φcan
Pose estimation network Φcat
Trainable function
Differentiable non-trainable function
Loss
Shared weights

Fig. 2: **Proposed framework**. (Left) shows the proposed self-supervision using pose consistency, and (Right) shows our compression pipeline from multiple networks to a single network by Conditional Instance Normalization (CIN) [7] and Knowledge Distillation (KD) [16]. The combined solution uses our object specific self-supervised networks as teachers in KD.

3 Proposed Methods

3.1 Overview

Our work largely consists of two parts: self-supervision and KD with parameterisation, based on the same pose estimator backbone. The two methods are then combined as shown in Fig. 2.

Self-supervision for lack of real annotated RGB images. We propose a self-supervised method which results in a 6D pose estimator via *pose consistency* without any form of pose ground truth on real RGB images. The proposed method adopts learning-by-synthesis in an end-to-end manner, via differentiable PnP and a neural renderer. In addition to pose consistency, silhouette masking, perceptual loss, occlusion augmentation and synthetic supervised loss are investigated for self-supervision. An ablation study is presented to investigate how different component methods contribute to accuracy.

KD with parameterisation for scalability. Given a DNN model, we consider adding more parameters to efficiently embed multiple objects to the model. For parameterisation, we propose primarily the use of Conditional Instance Normalization (CIN) [7] as a conditioning strategy, which can be efficiently applied with almost negligible overhead. The conditional information contains object identity, which is obtained by a state-of-the-art 2D object detector. We also apply Knowledge Distillation [16] and Hint Training [50] to help compress multiple single-object pose networks to a single network.

The combined solution. The two approaches above are put together. We first train multiple single-object pose estimators by self-supervision, then use them as teachers to distill knowledge to a single multi-object pose estimator conditioned by CIN.

Baseline pose estimator $\mathbf{\Phi}$. We choose BB8 [44] as the baseline 6D pose estimator for our experiment although any off-the-shelf pose estimator can be used. The simplest form is the direct estimation [27] of translation and the orientation in quaternion representation or Euler angles, however it has been shown that the direct estimation performs poorly. BB8 infers the coordinates of the 2D projection of the 3D bounding box which is used by PnP algorithm to recover translation and orientation. The pose estimator is light-weight and it has been used extensively for various purposes such as category-level object pose estimation [10] and feature mapping [44].

3.2 Self-supervised 6D Pose Learning

Given a labeled dataset in the source domain and an unlabeled dataset in the target domain, our goal is to conduct the learning in the source domain and to generalise to the target domain. We assume that the primary differences between the two domains, real and synthetic, are in low-level properties such as illumination and noise rather than high-level geometric variations [3]. Our model utilises higher-level commonality information such as geometry, shape, and texture as supervision to learn a pose estimator which generalises to the target domain. Let $\mathbf{X}^g = \{x_i^g, y_i^g, h_i^g | h_i^g = \pi(y_i^g, o, k)\}$ and $\mathbf{X}^r = \{x_i^r\}$ be the dataset in the source (synthetic) and target (real) domain respectively. $\pi(\cdot)$ is a function which projects the 3D bounding box of the object to the image plane given o, k and the object pose where o and k are object mesh and camera intrinsic parameters respectively. During training (see Fig.2, top), the baseline pose estimator infers projected 3D bounding box coordinates on a real RGB image, $h_1 = \Phi(x^r; \theta)$. The projected 3D bounding box coordinates h_1 is used to yield $R(P(h_1, k), o, k) = \{r_{h_1}, m_{h_1}, y_{h_1}^g\}$ via the PnP algorithm $P(\cdot)$ and the renderer $R(\cdot)$. The functional outputs are the new synthetic image r_{h_1}, and its silhouette m_{h_1} and pose label $y_{h_1}^g$. Masking function $M(x^r, r_{h_1}) = x_{h_1}^r$ outputs the masked real image using the silhouette. Finally the pose estimator predicts the bounding box coordinates for both the masked image $x_{h_1}^r$ and rendered synthetic image r_{h_1}, that is $\Phi(x_{h_1}^r; \theta) \rightarrow h_2$ and $\Phi(r_{h_1}; \theta) \rightarrow h_3$. Object poses are estimated multiple times across different domains, but the weights θ are shared.

Differentiable PnP and Neural Renderer. We use BB8 [44] as the baseline pose estimator. In order to make the framework end-to-end trainable, we need a differentiable component to convert the output of the pose estimator to the 6D pose. We use off-the-shelf implementations [31] and calculate the Jacobian matrix of derivatives of image points with respect to components of the 6D pose for differentiable PnP $P(\cdot)$. We use differentiable renderer [25], denoted as $R(\cdot)$, to generate synthetic images and silhouette given object mesh, camera parameters and object pose.

Pose consistency loss. The core of our idea is to encourage the pose estimator to estimate consistent poses for both synthetic and real domains. We define the pose consistency loss by penalising the discrepancy between the estimations from different domains. We use the Huber loss due to the robustness against the outliers:

$$L_{pose} = \mathbb{E}[\|(\Phi(x^r_{h_1}; \theta) - \Phi(r_{h_1}; \theta))\|_{hub}]. \tag{1}$$

Source domain supervised loss. The source domain supervised loss exploits the ground truth pose label that is available in the source domain:

$$L_{sup} = \mathbb{E}[\|\Phi(r_{h_1}; \theta) - \pi(y^g_{h_1}, o, k)\|_{hub}] + \mathbb{E}[\|\Phi(x^g; \theta) - h^g\|_{hub}], \tag{2}$$

where r_{h_1} and x^g are the rendered synthetic image via the proposed pipeline and the given source image. The synthetic supervised loss helps impose constraints on the real data self-supervision as learning by the pose consistency alone results in the singularity issue. For instance, the pose estimator can learn to always predict a trivial pose (e.g. frontal pose) regardless of the input data, in which the pose consistency loss is minimised. It also helps learn the prior geometric constraint such as the output of the pose estimator always needs to form a valid bounding box. In general, this loss narrows the gap between synthetic supervision and real self-supervision, preventing the model parameter learning from diverging.

Cross domain occlusion augmentation. To make the pose estimator robust to occlusions, randomly sized patches with Gaussian noise are placed on both the rendered image r_{h_1} and target domain image x^r in each iteration during training. The part of the silhouette m_{h_1} occluded by the noise patch is excluded to avoid the masked real image $x^r_{h_1}$ containing the noise patch.

Silhouette masking. The silhouette masking augments the input image by removing background with the silhouette rendered using the estimation h_1: $M(x^r, r_{h_1}) = x^r_{h_1}$, where $M(\cdot)$ is the alpha blending to remove the background given the silhouette. The masked image again is used to estimate the pose h_2 which is encouraged to be consistent with h_1, which is obtained without masking as $L_{pose2} = \mathbb{E}[\|\Phi(x^r; \theta) - \Phi(x^r_{h_1}; \theta)\|_{hub}]$. The intuition is that in order to achieve this goal, the estimated pose h_1 needs to be correct, otherwise the foreground image would be removed. This results in the pose estimator not being able to estimate h_2 consistently.

Overall training objective. In addition to the expected loss over the consistency and the source domain supervised loss, we also use an additional perceptual distance [71] to encourage the rendered image r_{h_1} and the masked real image

$x^t_{h_1}$ to be perceptually similar. We compute the distance in the feature space of pre-trained VGG [53] network between the two images, similar to [21]. The overall training objective is given as:

$$L_{tot} = \lambda_1 L_{pose} + \lambda_2 L_{pose2} + \lambda_3 L_{sup} + \lambda_4 L_{percep}, \tag{3}$$

where λ_i with $i = 1...4$ are weights that control the interaction of the losses. With the above objective, the proposed framework allows us to learn the pose estimator Φ without any form of labels in the target domain (i.e. real RGB images). At the test time, only Φ is used which is light-weight and fast.

3.3 Scaling to Multiple Objects

In this section, we investigate different techniques to bridge the performance gap between single and multi-object pose estimators.

Conditioning a Multi-Object Pose Estimator. Conventional pose estimators either have a single network per object, or a network that has multiple outputs for multiple objects as shown in Fig. 2. Our aim is to condition the network to increase its capacity for multiple object pose estimation. We design a new object conditioning strategy built on top of CIN [7]. A feature map F is conditionally normalised by:

$$cin(F, c) = \gamma_c \left(\frac{F - \mu}{\sigma} \right) + \beta_c, \tag{4}$$

where μ and σ are the channel-wise feature map's mean and standard deviation, and γ_c and β_c are the class's c scalar and bias parameters. Although originally formulated for style transfer purposes [7], we found that CIN can be successfully applied on multi-object 6D pose estimation tasks as shown in the experimental section. When CIN is applied, we observe that the multi-output models performed on par with the single-output models. Therefore, we further reduce the existing overhead by pairing every object's bounding box keypoints [2], which replaces the multi-output formulation with a single output.

Guided optimization with Knowledge Distillation and Hint Training. We examine Knowledge Distillation (KD) for compressing multiple single-object networks to a single multi-object network. While KD is typically applied to classification tasks [16], applying it to regression problems is less straightforward. Efforts in optimizing KD in this context have shown that a loss formulation needs to be carefully constructed [52,5]. In our scenario, the upper-bounded KD loss [5] is defined as:

$$L_{KD}(h_s, h_t, h_{gt}) = \begin{cases} ||h_s - h_{gt}||^2, \text{ if } ||h_s - h_{gt}||^2 < |h_t - h_{gt}||^2 \\ ||h_s - h_t||^2 + ||h_s - h_{gt}||^2, \text{ otherwise} \end{cases}, \tag{5}$$

where h_s and h_t are the student and teacher output and h_{gt} is the ground truth 3D bounding box label. Another useful component of the distillation process is Hint Training (HT) [50]. A hint is a teacher's intermediate feature map which serves as student guidance during training. In our approach, HT is used as an additional regularization mechanism which allows our multi-object student to better capture the teacher's knowledge. For this purpose, we pick the output of the last convolutional layer of Φ_t as our hint. In [50], the hint loss is defined by the Euclidean distance between feature maps. An adaptation layer is added if both feature maps are not of the same size. In line with [5], we found the adaptation layer to be crucial to effectively train the network using hints even if the feature maps have identical sizes. Our adapted hint training loss is:

$$L_{HT}(F_t, F_s) = \|\Psi_t(F_t; \theta_t) - F_s\|^2, \tag{6}$$

where Ψ_t is a 1x1 convolutional layer with learnable parameters θ_t. F_t and F_s are the feature maps of teacher and student respectively.

3.4 The Combined Solution

We combine both approaches by making use of multiple single object models trained with self-supervision as specialist teachers in our distillation framework. In our proposal, the knowledge distilled to the student does not come from a larger teacher but rather from multiple object specific teachers. During training, in order to produce the teacher's output h_t and hint F_t, we select the teacher according to the training instance class, i.e. object. In fact, our task becomes more challenging than the previous case due to the absence or lack of labelled real data. We design our self-supervised KD loss L_{KD^*} by modifying the upper-bounded loss taking into account the unavailability of real image labels:

$$L_{KD^*}(h_s, h_t, h_{gt}) = \begin{cases} L_{KD}(h_s, h_t, h_{gt}), \text{ if } h \in \mathbf{X}^g \\ ||h_s - h_t||^2, \text{ otherwise} \end{cases} \tag{7}$$

In practice, when distilling from the self-supervised teachers, we use the standard upper-bound loss L_{KD} when the instance is synthetic, otherwise we supervise the student with the teacher's output. In order to optimize our multi-object combined solution, the loss function is defined as $L_{CS} = L_{KD^*} + L_{HT}$.

4 Evaluation

We evaluate the efficacy of our methods through experiments, and ablation studies are presented to investigate how different component methods contribute to the method's accuracy. The proposed self-supervised method delivers significantly higher accuracies compared to synthetic only baselines, and almost as good as fully supervised methods using real data. With our conditioning along with knowledge distillation and hint training, we are able to generate highly optimised multi-object pose estimation models which are considerably faster and smaller than their combined single model counterparts with comparable accuracy performance. We present our results on the commonly used benchmark LINEMOD, LINEMOD OCCLUSION, containing 13 and 8 objects respectively, as well as the more challenging 30 object T-LESS dataset.

Metrics. All reported results are percentage of correctly predicted poses (i.e.

Table 1: Results on LINEMOD using ADD metric. Results for SSD6D and AAE are from [57].

Training Type	w/o real image (Domain randomization)			w/ real image w/o labels (Unsupervised Domain Adaptation)			w/ real images and labels
Method / **Object**	BB8 [44]	SSD6D [26]	AAE [57]	BB8 [44] +GRL [8]	BB8 [44] +PixelDA [3]	BB8 [44]+ OURS	BB8 [44]
Ape	0.46	0.00	3.96	0.74	1.40	20.98	24.79
Benchvise	0.75	0.18	20.92	3.00	0.66	65.97	63.07
Cam	0.00	0.41	30.47	0.56	1.02	39.20	46.43
Can	0.00	1.35	35.87	0.96	0.10	54.50	61.61
Cat	0.00	0.51	17.90	4.52	0.66	64.12	46.39
Driller	0.00	2.58	23.99	1.26	0.19	68.05	55.86
Duck	0.09	0.00	4.86	5.98	0.18	26.75	27.11
Eggbox	0.00	8.90	81.01	29.39	0.00	69.15	88.62
Glue	0.38	0.00	45.49	8.65	11.85	71.59	84.29
Holepuncher	0.00	0.30	17.60	2.64	0.91	19.09	34.27
Iron	0.10	8.86	32.03	12.10	3.60	71.90	76.90
Lamp	0.00	8.2	60.47	0.93	1.39	67.68	79.16
Phone	0.00	0.18	33.79	2.95	4.00	47.83	55.48
Mean	0.14	2.42	28.65	5.67	1.98	52.83	57.23

Table 2: Ablation study showing the impact of the combined solutions using ADD metric on LINEMOD dataset.

Method / **Object**	Multi+KD*	CIN+KD*	Multi+KD*+HT	CIN+KD*+HT	Single
Ape	12.53	14.48	21.86	16.62	20.98
Benchvise	25.86	43.11	30.64	46.20	65.97
Cam	17.61	10.75	20.39	18.54	39.20
Can	31.67	31.95	30.33	40.02	54.50
Cat	30.70	22.50	27.31	43.22	64.12
Driller	27.49	37.66	34.08	49.08	68.05
Duck	13.87	16.59	18.95	12.60	26.75
Eggbox	48.77	43.33	52.23	42.22	69.15
Glue	35.00	37.05	37.47	53.90	71.59
Holepuncher	11.55	16.64	16.18	10.54	19.09
Iron	41.10	38.40	45.30	48.00	71.90
Lamp	34.81	35.56	35.19	43.98	67.68
Phone	18.06	21.11	22.30	29.49	47.83
Mean	26.85	28.39	30.17	34.95	52.83

object recall rate) by one of the following metrics. We use 3 commonly used metrics with standard parameter settings to determine whether given predictions are correct. The following thresholds are used for each metrics: for 2D projection [4] we use 5 pixel threshold; 10% of the object's diameter for AD{D|I} [15]; and $err_{vsd} < 0.3$ with tolerance $\tau = 20mm$ and $\delta = 15mm$ only considering objects with visibility higher than 10% for VSD [17] following the protocols in [57]. Benchmark evaluation toolkit for the challenge [18] is used with default setting for evaluation of Visual Surface Discrepancy (VSD).

4.1 Implementation Details

Self-supervised 6D pose learning. For training, a batch size of 16 is used for each iteration with the ADAM optimiser [28]. The initial learning rate is 1e-5 for the first 15 epochs and reduced to 1e-6 for the next 10 epochs. Weights in Eq. 3 are $\lambda_1 = \lambda_2 = \lambda_3 = 1$ and $\lambda_4 = 20$. For the LINEMOD dataset, following the standard practice, 15% of the images are used *without* labels for training and the rest for testing. The selection of training images follows the strategy in [4]. The T-LESS dataset provides real images for training and again we use them without labels. To be robust against inaccurate detection, we randomly perturb center position and rescale the bounding box during training. We initialise the network by training with synthetic data only where the early layers are frozen to avoid overfitting to the synthetic dataset. Then all parameters are updated after initialisation. For all testing, Faster-RCNN [48] is used to detect objects unless stated otherwise. More details on parameters are in the supplementary.

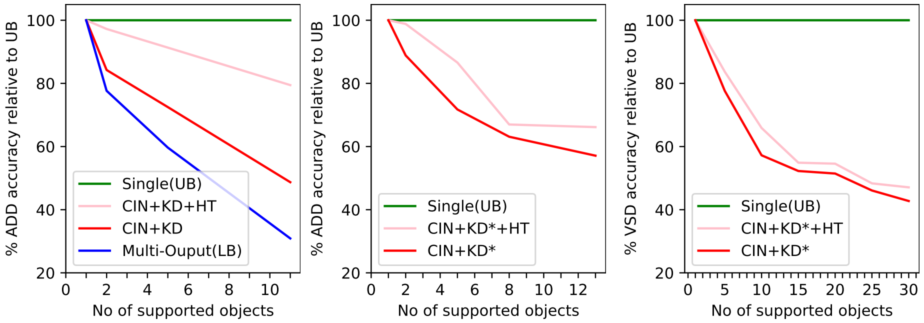
Scaling up to multiple objects. Dataset and training strategy follows the self-supervised framework except batch size is increased to 64 and learning rate is increased to 0.001.

4.2 Results on Self-supervision

Ablation Study Table 3 shows the influence of different components for the self-supervised object pose estimation on camera object from the LINEMOD dataset. It can be observed that silhouette masking is the

Table 3: Study of how different loss and augmentations affect the performance on the camera object in the LINEMOD dataset.

Synthetic Supervision	Silhouette masking	Perception Loss	Occlusion	ADD
✓				0.00
✓	✓			29.66
✓	✓	✓		35.31
✓	✓	✓	✓	39.20



(a) (b) (c)

Fig. 3: **Ablation Study.** (a) Relative performance comparing the single model (upper bound, UB) to a naive extension to multi object (lower bound, LB) and our multi-object proposed methods, trained on different number of objects with real annotated data. (b) Relative performance of our combined solution, trained with self-supervised single-object as teachers and synthetic data on LINEMOD and (c) on T-LESS datasets. Best viewed in colour.

Table 4: Results of our self-supervised approach on the T-LESS dataset.

Object Id / **Method**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
AAE [57] w/ GT 2D BBs	12.33	11.23	13.11	12.71	66.70	**52.30**	36.58	22.05	46.49	14.31	15.01	31.34	13.60	45.32	50.00	
Ours	**48.90**	**40.76**	**73.37**	**44.86**	**77.45**	44.99	**45.04**	**39.35**	**74.47**	**67.16**	**42.53**	**53.98**	**76.78**	**48.77**	**97.81**	
Object	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Mean
AAE [57] w/ GT 2D BBs	36.09	81.11	**52.62**	**50.75**	**37.75**	**50.89**	**47.60**	**35.18**	11.24	37.12	28.33	**21.86**	**42.58**	**57.01**	**70.42**	36.79
Ours	**95.68**	**96.08**	33.24	22.52	17.16	30.74	34.24	25.74	**35.35**	**37.36**	**31.94**	11.70	31.98	29.82	52.72	**48.75**

most critical component of the framework. Silhouette masking imposes a strong constraint as it is very difficult to satisfy the consistency criterion if the first estimation h_1 is wrong. This intuition is further reinforced by the experiments. Table 1 shows that for the objects with more distinctive silhouette for different poses (e.g. Benchvise) the performance is better than in round objects (e.g. Ape). Additionally, we found that the perception loss helps to make the training stage more stable and the occlusion augmentation further boosts the performance.

6D Object Pose Estimation Table 1 provides a comparison of the results obtained by different domain adaptation methods without real data annotation using ADD metric [15], as well as the baseline pose estimator trained with real data and pose labels which serve as upper bound. Note that we use our own implementation of BB8 [44] which achieves a better performance than

Table 5: Results of domain adaptation methods on LINEMOD using the 2D projection metric. Results except for our method are from [45].

Training Type	w/ real image w/o labels	w/ real images w/ labels			
Method	Ours	[51]	GRL [8]	DDC [61]	Feature Mapping [45]
Mean	90.69	90.0	90.8	88.0	93.4

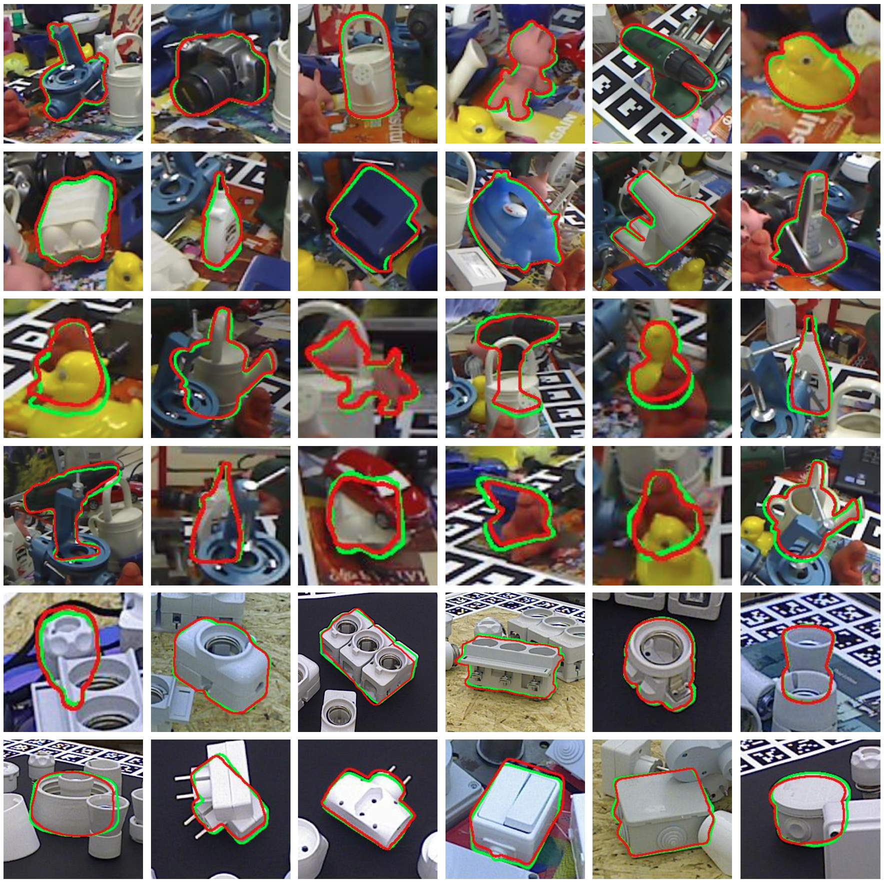


Fig. 4: **Qualitative results on all datasets.** LINEMOD (rows 1-2), LINEMOD OCCLUSION (rows 3-4), T-LESS (rows 5-6). Green and red silhouette correspond to ground truth and prediction respectively. Best viewed in colour.

the one reported in the original work. Our detector provides accurate bounding boxes with an accuracy of 99.24% and 97.29% with 50% and 75% IOU respectively, for all objects in the LINEMOD dataset. [44,26] trained only with synthetic images generalise poorly to real test images, as also observed in [69,57]. Although AAE [57] randomises various parameters, variability in augmentation for synthetic images does not generalises well to the real images. For unsupervised domain adaptation, although PixelDA [3] generates perceptually realistic images, they do not contribute much for the network to learn domain invariant features as also reported in [45]. This can be explained by the fact that the number of real images allowed in the standard LINEMOD protocol (15%) may not be enough to learn the pixel distribution. Using GRL [8] without real label only slightly improves the generalisability from synthetic only training as the only supervision is from the domain label disregarding the task at hand. Our method significantly outperforms all methods and almost reaches the performance of the fully supervised methods with real labels.

In Table 5, we use the 2D projection metric to compare more diverse state-of-the-art methods. All results are obtained with ground truth 2D object center and bounding box as in [45]. Our method shows competitive results without using any form of pose ground truth. Table 4 provides the comparison with AAE [57] using VSD metric on T-LESS dataset. For our method, full 6D object poses including translations are inferred from the output of the network, whereas AAE [57] infers only orientation and translations are calculated using projective distance estimation. Our method performs better on most objects. Low performance on objects 19-23 can be explained by the fact that the provided reconstructed mesh is missing parts (i.e. pins) causing rendered images to be substantially different from the real images. This violates our assumption that the difference between the two domains should be low-level, not geometric variations as stated in Section 3.2. Results show that our method outperforms the baseline even in a highly cluttered environment with various symmetrical objects.

Fig. 4 visualises some successful and unsuccessful estimations on all datasets. Even with a mild partial occlusion, our proposed method trained without real labels can accurately estimate poses.

Table 6: Results of our combined solution on LINEMOD OCCLUSION using the ADD metric. Results of the evaluated methods are presented in [70].

Training Type	w/ real images and labels					w/ real image but w/o labels
Method	YOLO6D [58]	PoseCNN [66]	SSD6D+Ref [35]	HMap [38]	DPOD [70]	Combined Solution
Mean	6.42	24.9	27.5	30.4	32.79	15.35

4.3 Results on Multi-object Pose Estimation

We first evaluate our proposed methods for multi-objects with full supervision. Using a smaller baseline model, we perform an ablation study by measuring our methods' performance on varying numbers of LINEMOD objects. The Upper Bound (UB) is defined as the average accuracy of the single-object models while the lower bound (LB) refers to a multi-object model with the standard multi-output formulation. Fig. 3a shows the change in the accuracy of our proposed methods relative to the UB. When applying CIN conditioning, the models learned object specific features more easily, resulting in an improvement over LB for all different numbers of objects. We further boost the performance by applying KD+HT, where we distill the knowledge from multiple teachers specialised in each object.

In our experiments using 11 LINEMOD objects, we also compared one-hot vector and CIN conditioning: both delivered similar accuracy and yielded similar parameter reduction of 90.7%. With our implementation of method [47] however, the reduction was only 80%.

4.4 The Combined Solution

We combine both methods and present a self-supervised multi-object pose estimator. Due to the lack of ground truth annotation for real images, our model is optimized with L_{KD^*} as defined in Section 3.3. The LB multi-output model should be trained on synthetic data only which has been previously shown to perform very low in Table 1. Therefore, in this ablation study, we omit LB. In Fig. 3b, we report the relative performance of our combined solution relative to UB. Training with KD* and HT achieves the best performance which is in line with Fig. 3a. This shows that aligning features between the student and the multiple teachers is important in compressing models. A more detailed analysis of the combined solution on the self-supervision framework can be found in Table 2. The mean accuracy across all objects increases with the addition of different components. We compressed 13 object models into a single conditioned model, reducing the total number of parameters by 92.2% while compromising only 35% accuracy performance. This is a significant reduction compared with over 50% accuracy loss measured on the multi-output baseline with the same parameters.

Table 7: **Ablation study:** Different model capacities on the T-LESS dataset, reported with the VSD metric. The numbers in the bracket indicate the number of teacher models compressed into each student model.

Compression	CIN	CIN
Rate	KD*	KD*+HT
97% (30)	20.84	22.95
93% (15)	25.94	27.08
90% (10)	34.82	37.02
80% (5)	36.36	39.12

Table 6 shows our combined solution performance on LINEMOD OCCLUSION and compares with state-of-the-art 6D pose estimators trained with real labels. The result shows that our proposed self-supervised method, in combination with KD*+HT and CIN conditioning, is not only able to learn robust features against occlusion in the absence of labels but is also able to outperform prior multi-object works such as [58].

In Fig. 3c, we show how the performance changes when we incrementally increase the number of T-LESS objects. The objects are in the order they appear on T-LESS. The figure indicates the relative VSD accuracy compared to the UB. For various number of objects, we observe incremental improvements when applying HT.

Additionally, we increase the capacity of our 30 object student in order to evaluate how the student performance scales w.r.t. model capacity. In practice, we trained multiple student models, each trained for different objects and without any overlap. We compared these combined results to the 30 object model. In Table 7, we compare multiple compression rates. In comparison to AAE [57], our model achieves better accuracies at 90% compression rate (10 objects per model) and below, as shown in Table 7.

5 Conclusion

This paper addresses the two challenges of 6D object pose estimation. Existing methods perform poorly when real data annotation is not available and/or do not scale up to multiple objects. The proposed self-supervised learning framework can learn an object pose estimator given images in the target domain without any form of annotation and almost reaches the performance of fully supervised training. We also perform model compression with CIN and distillation techniques to achieve high performance multi-object pose estimator with minimal increase in the model capacity. Our evaluations show the effectiveness of the proposed framework on multiple datasets. Ablation study for self-supervision shows that silhouette masking augmentation is crucial to impose pose consistency. Currently unlabelled real image used for training does not include occlusion. A possible future work could extend the proposed method to learn pose estimation from unlabelled images with occlusion along with the foreground prediction. Moreover, given the efficacy of our framework, an important direction to follow could be to extend our work with more baselines using different modalities such as point clouds and depth for self-supervised multi-object pose estimation.

References

1. Belagiannis, V., Farshad, A., Galasso, F.: Adversarial network compression. In: ECCV (2018)
2. Bilen, H., Vedaldi, A.: Universal representations: The missing link between faces, text, planktons, and cat breeds. arXiv preprint arXiv:1701.07275 (2017)
3. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: CVPR (2017)
4. Brachmann, E., Michel, F., Krull, A., Ying Yang, M., Gumhold, S., et al.: Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In: CVPR (2016)
5. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: NIPS (2017)
6. Chen, T., Zhai, X., Ritter, M., Lucic, M., Houlsby, N.: Self-supervised gans via auxiliary rotation loss. In: CVPR (2019)
7. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. arXiv preprint arXiv:1610.07629 (2016)
8. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. JMLR (2016)
9. Georgakis, G., Karanam, S., Wu, Z., Kosecka, J.: Matching rgb images to cad models for object pose estimation. arXiv preprint arXiv:1811.07249 (2018)
10. Grabner, A., Roth, P.M., Lepetit, V.: 3d pose estimation and 3d model retrieval for objects in the wild. In: CVPR (2018)
11. Gupta, S., Agrawal, A., Gopalakrishnan, K., Narayanan, P.: Deep learning with limited numerical precision. In: ICML (2015)
12. Han, S., Pool, J., Narang, S., Mao, H., Gong, E., Tang, S., Elsen, E., Vajda, P., Paluri, M., Tran, J., Catanzaro, B., Dally, W.J.: Dsd: dense-sparse-dense training for deep neural networks. In: ICLR (2017)
13. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. In: NIPS (2015)
14. He, K., Gkioxari, G., Dollr, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
15. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: ACCV (2012)
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: Deep Learning and Representation Learning Workshop, NIPS (2015)
17. Hodaň, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M., Zabulis, X.: T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. WACV (2017)
18. Hodaň, T., Michel, F., Brachmann, E., Kehl, W., Glent Buch, A., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X., Sahin, C., Manhardt, F., Tombari, F., Kim, T.K., Matas, J., Rother, C.: Bop: Benchmark for 6d object pose estimation. ECCV (2018)
19. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. arXiv preprint arXiv:1711.03213 (2017)
20. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV (2017)
21. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: ECCV (2018)

22. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks. In: NIPS (2016)
23. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. TPAMI (2014)
24. Kaskman, R., Zakharov, S., Shugurov, I., Ilic, S.: Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (Oct 2019)
25. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: CVPR (2018)
26. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In: ICCV (2017)
27. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: ICCV (2015)
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
29. Kulkarni, N., Gupta, A., Tulsiani, S.: Canonical surface mapping via geometric cycle consistency. In: ICCV (2019)
30. Lee, S., Kim, D., Kim, N., Jeong, S.G.: Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In: ICCV (2019)
31. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate o (n) solution to the pnp problem. IJCV (2009)
32. Li, Z., Wang, G., Ji, X.: Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In: ICCV (2019)
33. Liu, X., He, P., Chen, W., Gao, J.: Multi-task deep neural networks for natural language understanding. In: ACL (2019)
34. Luo, J.H., Wu, J., Lin, W.: Thinet: A filter level pruning method for deep neural network compression. In: ICCV (2017)
35. Manhardt, F., Kehl, W., Navab, N., Tombari, F.: Deep model-based 6d pose refinement in rgb. In: ECCV (2018)
36. Mees, O., Tatarchenko, M., Brox, T., Burgard, W.: Self-supervised 3d shape and viewpoint estimation from single images for robotics. In: IROS (2019)
37. Murez, Z., Kolouri, S., Kriegman, D., Ramamoorthi, R., Kim, K.: Image to image translation for domain adaptation. In: CVPR (2018)
38. Oberweger, M., Rad, M., Lepetit, V.: Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In: ECCV (2018)
39. Palazzi, A., Bergamini, L., Calderara, S., Cucchiara, R.: End-to-end 6-dof object pose estimation through differentiable rasterization. In: ECCV (2018)
40. Park, K., Patten, T., Vincze, M.: Pix2pose: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In: ICCV (2019)
41. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: NIPS Autodiff Workshop (2017)
42. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: Pvnet: Pixel-wise voting network for 6dof pose estimation. In: CVPR (2019)
43. Polino, A., Pascanu, R., Alistarh, D.: Model compression via distillation and quantization. In: ICLR (2018)
44. Rad, M., Lepetit, V.: Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In: ICCV (2017)
45. Rad, M., Oberweger, M., Lepetit, V.: Feature mapping for learning fast and accurate 3d pose inference from synthetic images. In: CVPR (2018)

46. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Learning multiple visual domains with residual adapters. In: NIPS (2017)
47. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Efficient parametrization of multi-domain deep neural networks. In: CVPR (2018)
48. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS (2015)
49. Riza Alp Guler, Natalia Neverova, I.K.: Densepose: Dense human pose estimation in the wild. In: CVPR (2018)
50. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. In: ICLR (2015)
51. Rozantsev, A., Salzmann, M., Fua, P.: Beyond sharing weights for deep domain adaptation. TPAMI (2018)
52. Saputra, M.R.U., de Gusmao, P.P., Almalioglu, Y., Markham, A., Trigoni, N.: Distilling knowledge from a deep pose regressor network. In: ICCV (2019)
53. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
54. Sock, J., Kim, K.I., Sahin, C., Kim, T.K.: Multi-task deep networks for depth-based 6d object pose and joint registration in crowd scenarios. In: BMVC (2018)
55. Sofiiuk, K., Barinova, O., Konushin, A.: Adaptis: Adaptive instance selection network. In: ICCV (2019)
56. Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., Freeman, W.T.: Pix3d: Dataset and methods for single-image 3d shape modeling. In: CVPR (2018)
57. Sundermeyer, M., Marton, Z.C., Durner, M., Brucker, M., Triebel, R.: Implicit 3d orientation learning for 6d object detection from rgb images. In: ECCV (2018)
58. Tekin, B., Sinha, S.N., Fua, P.: Real-time seamless single shot 6d object pose prediction. In: CVPR (2018)
59. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: IROS (2017)
60. Tran, N.T., Tran, V.H., Nguyen, N.B., Cheung, N.M.: An improved self-supervised gan via adversarial training. arXiv preprint arXiv:1905.05469 (2019)
61. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)
62. Vanhoucke, V., Senior, A., Mao, M.Z.: Improving the speed of neural networks on cpus. In: Deep Learning and Unsupervised Feature Learning Workshop, NIPS (2011)
63. Wan, C., Probst, T., Gool, L.V., Yao, A.: Self-supervised 3d hand pose estimation through training by fitting. In: CVPR (2019)
64. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6d object pose and size estimation. In: CVPR (2019)
65. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: NIPS (2016)
66. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv preprint arXiv:1711.00199 (2017)
67. Yuan, S., Ye, Q., Stenger, B., Jain, S., Kim, T.K.: Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In: CVPR (2017)
68. Zakharov, E., Shysheya, A., Burkov, E., Lempitsky, V.: Few-shot adversarial learning of realistic neural talking head models. arXiv preprint arXiv:1905.08233 (2019)

69. Zakharov, S., Kehl, W., Ilic, S.: Deceptionnet: Network-driven domain randomization. In: ICCV (2019)
70. Zakharov, S., Shugurov, I., Ilic, S.: Dpod: 6d pose object detector and refiner. In: ICCV (2019)
71. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)

Supplementary Material

This supplementary material provides details on network design and additional results and examples. Some contents from the main paper are reproduced so that this document is self-contained.

1 Implementation Details

Pytorch [41] is used to train and test the proposed method on a machine with Intel i5 and GTX1080TI. The meshes used for the differentiable renderer [25] were downsampled to reduce the training time if the number of faces are too large.

1.1 Conditioning

We normalize the output of last convolutional layer of the BB8 architecture, which has a channel size of 512. We design a lookup table containing the learnable CIN parameters [7]. For each object supported by the multi-object model, this table contains scaling and bias normalization parameters for every feature map channel. In practice, for a multi-object model trained on the entirety of the TLESS dataset, i.e. 30 objects, the lookup table has a size of 512x30x2. Given an input pair of a 2D RGB image of an object and its correspondent object class, we compute the forward and backward pass with the correct selection of CIN parameters. Due to the nature of the selection operation, it is possible to perform batch training.

1.2 Knowledge Distillation and Hint Training

In our Knowledge Distillation (KD) [16] and Hint Training (HT) [50], we use the identical network architecture for the student and teachers, except the students' output layer and/or conditioning parameters. Therefore, we use the last convolutional layer before the CIN conditioning step as hint (the last red convolutional layer in Figure 1). Following [5,52], we use 1x1 convolutional adapters introduced in [50], which is applied to each teacher. Although the adapters increase the capacity of the pipeline, these are used only for Hint training. At inference time, they are not needed.

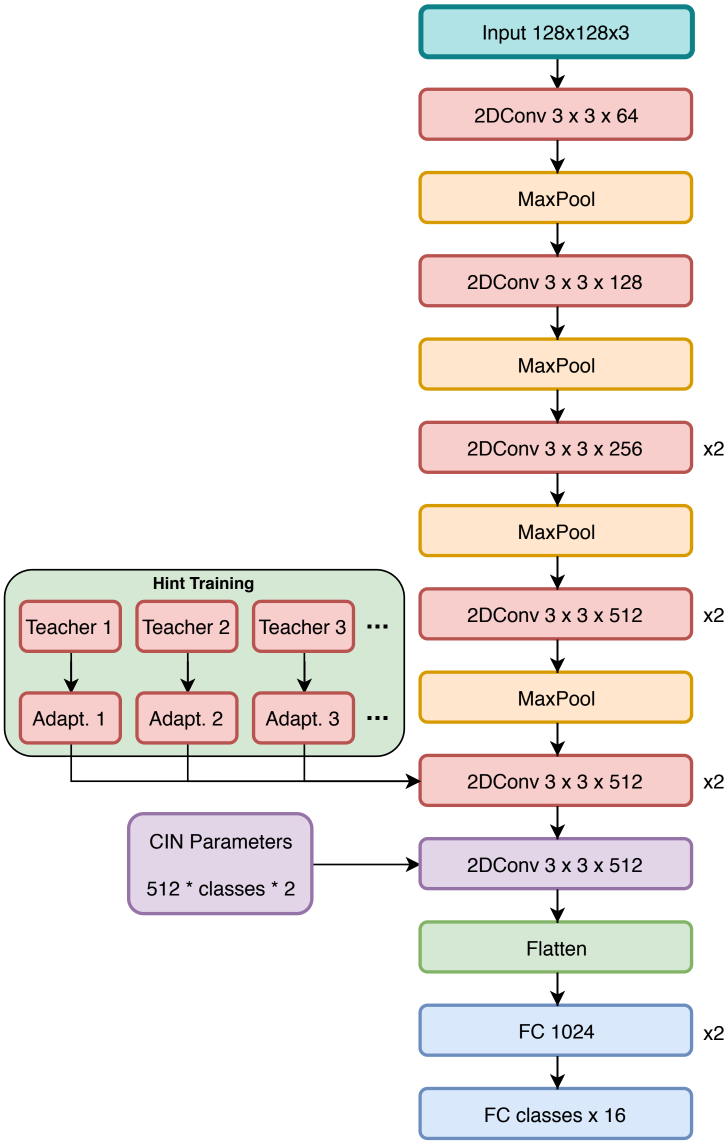


Fig. 1: Detailed architecture for the student-teacher knowledge distillation