SO-Pose: Exploiting Self-Occlusion for Direct 6D Pose Estimation

Yan Di [*1], Fabian Manhardt[2], Gu Wang[3], Xiangyang Ji[3], Nassir Navab[1] and Federico Tombari[1,2]

[1]Technical University of Munich, [2]Google, [3]Tsinghua University

[*]shangbuhuan13@gmail.com, fabianmanhardt@google.com, nassir.navab@tum.de,
wangg16@mails.tsinghua.edu.cn, xyji@tsinghua.edu.cn, tombari@in.tum.de

Abstract

Directly regressing all 6 degrees-of-freedom (6DoF) for the object pose (i.e. the 3D rotation and translation) in a cluttered environment from a single RGB image is a challenging problem. While end-to-end methods have recently demonstrated promising results at high efficiency, they are still inferior when compared with elaborate PnP/RANSAC-based approaches in terms of pose accuracy. In this work, we address this shortcoming by means of a novel reasoning about self-occlusion, in order to establish a two-layer representation for 3D objects which considerably enhances the accuracy of end-to-end 6D pose estimation. Our framework, named SO-Pose, takes a single RGB image as input and respectively generates 2D-3D correspondences as well as self-occlusion information harnessing a shared encoder and two separate decoders. Both outputs are then fused to directly regress the 6DoF pose parameters. Incorporating cross-layer consistencies that align correspondences, self-occlusion and 6D pose, we can further improve accuracy and robustness, surpassing or rivaling all other state-of-the-art approaches on various challenging datasets.

1. Introduction

Estimating the 6D pose has been widely adopted as an essential cue in high-level computer vision tasks, including robotic grasping and planning [1], augmented reality [39], and autonomous driving [25, 29]. Driven by the recent success of deep learning, current methods are capable of estimating the 6D pose in a cluttered environment at impressive accuracy and high efficiency [15, 47, 18]. Almost all current top-performing frameworks adopt a two-stage strategy that resorts to first establishing 2D-3D correspondences and then computing the 6D pose with a RANSAC-based Perspective-n-Point (PnP) algorithm [47, 20, 9]. Neverthe-

---

[*]Codes will be released at https://github.com/shangbuhuan13/SO-Pose

(a) Basic structure of baseline methods [12, 43]
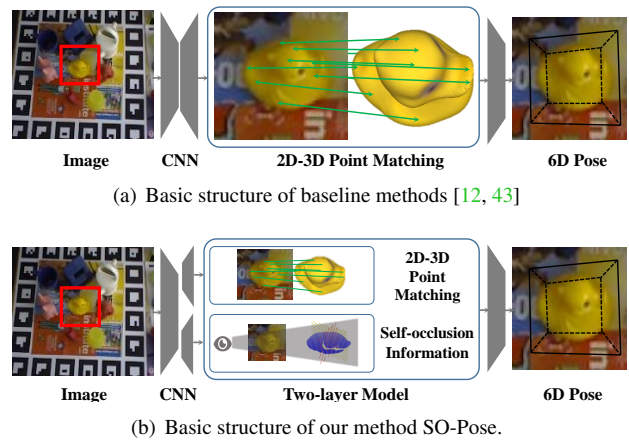

(b) Basic structure of our method SO-Pose.

Figure 1. **The basic structures of end-to-end 6D pose estimation methods.** Compared to single-layer methods [12, 43] that depend on 2D-3D point matching as intermediate results, our method SO-Pose presents a novel two-layer representation that additionally incorporates self-occlusion information about the object.

less, while achieving great results, these methods cannot be trained in an end-to-end manner and require extra computation for optimization of pose. Moreover, adopting surrogate training losses instead of directly predicting 6D poses also prevents further differentiable processing/learning (*e.g.* by means of self-supervised learning [42]) and does not allow to incorporate other down-stream tasks.

Despite two-stage approaches dominating the field, a few methods conducting end-to-end 6D pose estimation have been also recently proposed [12, 4, 42, 43]. They typically learn the 6D pose directly from dense correspondence-based intermediate geometric representations, as shown in Fig. 1(a). Nevertheless, although end-to-end methods keep constantly improving, they are still far inferior to two-stage methods harnessing multi-view consistency check [18], symmetry analysis [9], or disentangled predictions [20].

What limits the accuracy of end-to-end methods? After in-depth investigation in challenging scenes, we observe

that while the network is approaching the optimum, due to the inherent matching ambiguity of textureless object surface, mis-matching error caused by noise is inevitable, resulting often in one correspondence field corresponding to many 6D poses with similar fitting errors. This leads the training process to converge to a sub-optimum, hindering the overall 6D pose estimation performance. Since eliminating errors caused by noise is not trivial, an alternative solution to this problem is to replace the correspondence field with a more precise representation of the 3D object, thus reducing the influence of noise.

In this work, we attempt at closing the gap between end-to-end and two-stage approaches by leveraging self-occlusion information about the object. For an object in 3D space, we can logically only observe its visible parts due to the nature of the perspective projection. Yet, parts that are invisible due to (self-) occlusion are usually neglected during inference. Inspired by multi-layer models used in 3D reconstruction [34], we focus on self-occlusion information to establish a viewer-centered two-layer representation of the object pose. While the first layer preserves the correspondence field of visible points on the object and their projections, the second layer incorporates the self-occlusion information. In essence, instead of directly identifying whether and where each visible point occludes the object, we simplify the procedure by examining the self-occlusion between each pixel and the object coordinate planes. As illustrated in Fig. 3, the ray passing through the camera center and each visible point intersects the object coordinate plane at most three different locations. The coordinates of these intersections are then utilized to form the second layer representation of the object, as shown in Fig. 1(b). Finally, two cross-layer consistency losses are introduced to align self-occlusion, correspondence field and 6D pose simultaneously, reducing the influence of noise.

To summarize, our main contributions are as follows:

- We propose SO-Pose, a novel deep architecture that directly regresses the 6D pose from the two-layer representation of each 3D object.
- We propose to leverage self-occlusion and 2D-3D correspondences to establish a two-layer representation for each object in 3D space, which can be utilized to enforce two cross-layer consistencies.
- SO-Pose consistently surpasses all other end-to-end competitors on various challenging datasets. Moreover, SO-Pose also achieves comparable accuracy when compared with other state-of-the-art two-stage methods, whilst being much faster.

2. Related Works

The related works for monocular 6D pose estimation can be roughly partitioned into three different lines of works. In particular, while some methods directly regress the final 6D pose, others either learn a latent embedding for subsequent retrieval of the pose, or employ 2D-3D correspondences to solve for the 6D pose by means of the well-established RANSAC/PnP paradigm.

As for the first line of works, Kehl *et al.* [15] extends SSD [23] to estimate the 6D object pose, turning the regression into a classification problem. In their follow-up work [24], Manhardt *et al.* leverage multiple hypotheses to improve robustness towards ambiguities. In [26], the authors leverage ideas from projective contour alignment to estimate the pose. A few other works also make use of the point-matching loss to directly optimize for pose in 3D [45, 19, 18]. Finally, [12] and [43] both establish 2D-3D correspondences but attempt to learn the PnP paradigm in an end-to-end fashion.

The next branch employs latent embedding for pose estimation. These learned embeddings can be then leveraged for retrieval during inference. Specifically, inspired by [44, 16], Sundermeyer *et al.* [38] utilize an Augmented AutoEncoder (AAE) to learn a low-dimensional pose embedding. After localizing the object in image space using a 2D object detector [23, 21], the latent representation of the detection is calculate and compared against a pre-computed codebook to retrieve the pose. To further improve scalability to multiple objects, the authors of [37] propose to employ a single-shared encoder together with separate decoders for each object.

Finally, the last branch is grounded on establishing 2D-3D correspondences, before solving for the pose using RANSAC/PnP. Thereby, some works propose to regress the 2D projections of the 3D bounding box corners [30, 40]. To increase the robustness of these correspondences, Hu *et al.* predicts multiple hypotheses on the basis of segmented super-pixels. Nevertheless, most recent methods utilize 2D-3D correspondences with respect to the 3D model rather than the 3D bounding box. Peng *et al.* [28] demonstrate that keypoints away from the object surface induce larger errors and, therefore, instead sample several keypoints on the object model based on farthest point sampling. Hybrid-Pose [36] follows and develops [28] by introducing hybrid representations. Noteworthy, the majority of works within this branch, however, establishes dense 2D-3D correspondences [47, 20, 27, 9]. They are among the best-performing methods on several challenging benchmarks.

3. Methodology

Given an image I, we leverage a neural network to learn a mapping $f(\cdot)$ from I to the relative 3D rotation R and translation t, transforming the target object from the object frame to the camera frame,
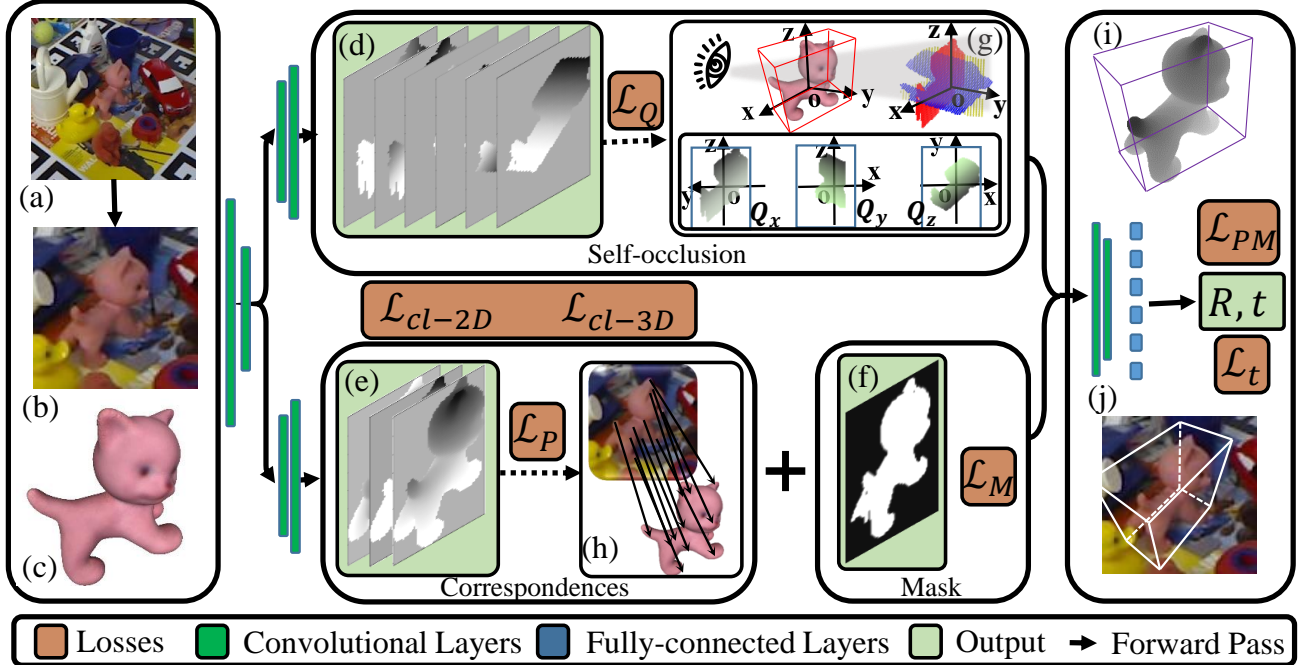
$$R, t = f(I; \Theta), \tag{1}$$

Figure 2. **Schematic overview of the proposed SO-Pose framework.** Given an input image (a) and 3D model (c), we first use an off-the-shelf object detector [31, 32] to crop the object of interest (b) from (a). Afterwards, (b) is fed into our encoder for high-level feature extraction. These features are then separately processed by two individual decoder networks to predict a two-layer representation. Thereby, while the first branch outputs a self-occlusion map (d), the latter branch estimates 2D-3D point correspondences (e) and object mask (f). In (h), we demonstrate the established 2D-3D correspondences between the visible surface of the object and the 3D model. A detailed illustration of self-occlusion is shown in (g). For a visible point on the object surface, it occludes coordinate planes $o-yz, o-xz, o-xy$ at Q_x, Q_y and Q_z. Finally, we feed the self-occlusion map (d), together with the 2D-3D point matching field (e) to the pose estimator block that predicts the final 6D pose. Exemplary output for depth map and the rendered bounding box using the estimated pose are shown in (i) and (j), respectively. $\mathcal{L}_*$ represent loss terms used in the training process.

with Θ denoting the trainable parameters of the utilized network. In a cluttered environment, the available object information is oftentimes severely limited due to (self-) occlusion. Moreover, directly regressing the 3D rotation parameters under occlusion has proven to be challenging [11]. Inspired by multi-layer models in 3D reconstruction [34], we propose to combine visible 2D-3D correspondences with invisible self-occlusion information to establish a two-layer representation for objects in 3D space, in an effort to capture more complete geometric features than single-layer approaches relying only on correspondences [12, 43]. Thereby, we enforce two cross-layer consistencies to align self-occlusion, correspondence field and 6D pose, to reduce the influence of noise and, thus, enhance pose estimation under various challenging external influences. The overall architecture of SO-Pose is illustrated in Figure 2.

3.1. Self-occlusion for Robust Pose Estimation

The vast majority of CNN-based 6D pose estimation approaches focus only on the visible part of an object while discard the occluded part [13, 9, 43]. Nevertheless, in complex environments, the visible region of an object is often-

times very limited or only exhibits little amount of textured information. Hence, single-layer representations can not encode the geometric features of the object completely and accurately, inducing ambiguities for 6D pose. Similar to multi-layer model in 3D reconstruction [34], we attempt to leverage self-occlusion information to obtain a richer representation of the 3D object. As shown in Fig. 2 (d) and (e), we combine self-occlusion with estimated 2D-3D correspondences to establish a novel two-layer representation for describing the pose of an object in 3D space. For better understanding, imagine a ray emitted from the camera center and passing through the object. This ray intersects the object surface at multiple different points, of which the first one is visible whereas all others are self-occluded. In contrast to [34] which records the coordinates of self-occluded points, we instead note the coordinates of the intersections between each ray and the coordinate planes of the object. As illustrated in Fig. 3, the ray OP intersects the object coordinate frames $o-yz, o-xz, o-xy$ at points Q_x, Q_y, Q_z. For an object obj, we combine $P, Q = \{Q_x, Q_y, Q_z\}$ to represent its two-layer model,

$$obj := \{P, Q\} \quad P \in V, \tag{2}$$

with V denoting the visible points of the current view w.r.t. the camera coordinate system. Notice that Q can be derived analytically from P, knowing the rotation R and translation t. Projecting P onto the 2D image plane, we obtain

$$\rho = \frac{1}{Z_P} KP, \tag{3}$$

with K describing the camera intrinsic matrix and $P = [X_P, Y_P, Z_P]^T$ denoting the visible 3D point. Further, the object coordinate plane w.r.t. the camera coordinate system is defined as

$$(Rn_*)^T X = (Rn_*)^T t. \tag{4}$$

where X represents a 3D point on the corresponding coordinate plane given

$$n_* = \begin{cases} n_x = [1,0,0]^T & X \in o - yz \\ n_y = [0,1,0]^T & X \in o - xz \\ n_z = [0,0,1]^T & X \in o - xy \end{cases} \tag{5}$$

From this we can derive Q_x, lying on the plane $o - yz$ intersected by the ray OP, as follows,

$$Q_x = \frac{(Rn_x)^T t}{(Rn_x)^T (K^{-1}\rho)} K^{-1}\rho \tag{6}$$

Substituting n_x with n_y or n_z in Eq. 6, we can respectively derive Q_y or Q_z.

Since P and Q are represented w.r.t. the camera coordinate system, their corresponding coordinates w.r.t. the object coordinate system are calculated as $P_0 = R^T P - R^T t$ and $Q_0 = R^T Q - R^T t$. Notice that we normalize P_0 and Q_0 based on the object diameter to stabilize optimization. Notably, as the ray passing through the camera center O and a visible point P may be parallel to one of the object coordinate planes, it can occur that the ray never intersects this plane. Therefore, to circumvent these cases and increase robustness, we only consider intersections inside the minimum bounding cuboid Ω of the object, as visualized in Fig. 3.

According to the definition of self-occlusion in Eq. 6, our two-layer representation exhibits 3 advantages over single-layer approaches. First, the self-occlusion coordinate Q is analytically derived with rotation and translation parameters for each visible point, independent of the object surface. Thus it eliminates errors arising from rendering. Moreover, since the self-occlusion coordinate Q_0 lies on a coordinate plane and has therefore only 2 degrees-of-freedom, we also only require to predict 2 values to represent Q_0. Hence, this acts as a regularization term which can reduce the influence of noise. Finally, since P and Q lie on the same line, we can derive several cross-layer consistencies which align self-occlusion, 2D-3D correspondences and 6D pose, increasing significantly accuracy and robustness of SO-Pose, especially in challenging environments.
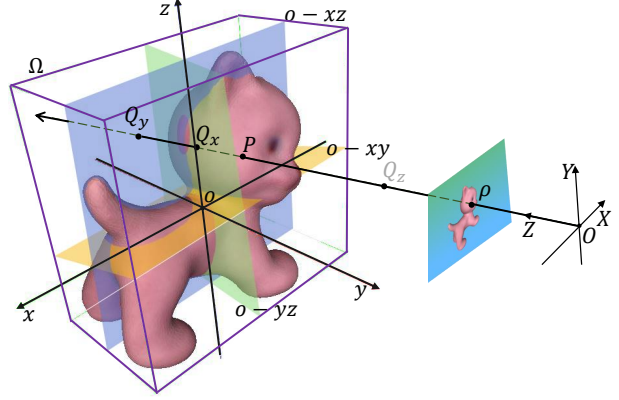

Figure 3. **Demonstration of self-occlusion.** The ray OP from camera center O towards the visible point P intersects the object coordinate planes $o - yz, o - xz, o - xy$ at 3 points Q_x, Q_y and Q_z. We only consider the points inside the pre-defined region Ω for training stability, thus Q_z would be removed in this example.

3.2. Cross-layer Consistency

Leveraging the estimated two-layer representation for objects in 3D space, we enforce two cross-layer consistency loss terms to jointly align self-occlusion, correspondence field and 6D pose parameters. When substituting Eq. 3 into Eq. 6 and rearranging, we obtain that

$$(Rn)^T tP = (Rn)^T PQ. \tag{7}$$

From this we enforce our first cross-layer consistency as

$$\mathcal{L}_{cl-3D} = \frac{1}{|Q_0|} \sum_{P \in V, Q_0 \in \Omega} \|(Rn)^T t(RP_0 + t) \\ - (Rn)^T (RP_0 + t)(RQ_0 + t)\|_1, \tag{8}$$

with $\| * \|_1$ representing the $L1$ loss and $|Q_0|$ denoting the number of intersections within Ω. Eq. 8 jointly aligns and refines 2D-3D correspondences P, self-occlusion Q and pose R, t in 3D space, based on the definition of Q in Eq. 6.

While the first cross-layer consistency is enforced in 3D space, we employ the second cross-layer loss on the 2D image plane. Since P and Q lie on the same ray, their projections describe the same point ρ on the image plane. As a consequence we can derive our 2D consistency term as follows,

$$\mathcal{L}_{cl-2D} = \frac{1}{|Q_0|} \sum_{P \in V, Q_0 \in \Omega} (\|e_{PQ}\|_1 + \|e_{Q\rho}\|_1) \tag{9}$$

with

$$e_{PQ} = \frac{1}{Z_Q} K(RQ_0 + t) - \frac{1}{Z_P} K(RP_0 + t) \tag{10}$$

and

$$e_{Q\rho} = \frac{1}{Z_Q} K(RQ_0 + t) - \rho. \tag{11}$$

where e_{PQ} forces P and Q to project onto the same 2D point, while $e_{Q\rho}$ forces Q to project to ρ, the corresponding ground truth projection of Q.

3.3. Overall Objective

SO-Pose takes a single RGB image as input and directly predicts 6D pose parameters of an object in 3D space. To establish a two-layer representation, our framework generates a correspondence field and three self-occlusion maps. In the following, both intermediate geometric features are concatenated and fed to the pose predictor to obtain the output pose in a fully differentiable fashion. Our overall objective function is composed of basic terms for pose, cross-layer consistency terms, and self-occlusion term,

$$\mathcal{L} = \mathcal{L}_{pose} + \mathcal{L}_{cl} + \mathcal{L}_{occ}, \tag{12}$$

with

$$\mathcal{L}_{cl} = \lambda_1 \mathcal{L}_{cl-2D} + \lambda_2 \mathcal{L}_{cl-3D} \tag{13}$$

and

$$\mathcal{L}_{occ} = \lambda_3 \mathcal{L}_Q. \tag{14}$$

In particular, $\mathcal{L}_{pose}$ is a combined loss term for correspondence field, translation parameters, visible mask, region classification and point matching as in [43]. We kindly refer the reader to [43] for more details on the pose terms.

As for self-occlusion, $\mathcal{L}_Q$ is composed of two parts,

$$\mathcal{L}_Q = \mathcal{L}_{Q_1} + \mathcal{L}_{Q_2}. \tag{15}$$

Thereby, we straightforwardly employ the L_1 loss according to

$$\mathcal{L}_{Q_1} = \frac{1}{|Q_0|} \sum_{Q_0 \in \Omega} \|Q_0 - \hat{Q}_0\|_1, \tag{16}$$

with $\hat{Q}_0$ denoting the ground truth self-occlusion coordinates.

Similarly, also for $\mathcal{L}_{Q_2}$ we directly employ the L_1 loss to ensure consistency after projection using

$$\mathcal{L}_{Q_2} = \frac{1}{|Q_0|} \sum_{Q_0 \in \Omega} \|\frac{1}{Z_Q} K(\hat{R} Q_0 + \hat{t}) - \rho\|_1. \tag{17}$$

Thereby, $\hat{R}$ and $\hat{t}$ represent the ground-truth rotation and translation. Eq. 17 enforces that all predicted self-occlusion coordinates Q_x, Q_y and Q_z reside on the same ray with respect to P.

4. Evaluation

In this section we compare SO-Pose to current state-of-the art methods in 6D pose estimation. We conduct extensive experiments on three challenging datasets to demonstrate the effectiveness and superiority of our approach. We also perform various ablation studies to verify that our two-layer model consistently surpasses the single-layer competitors.

4.1. Network Structure

We feed SO-Pose with a zoomed-in RGB image [20, 43] of size 256×256 as input and directly output 6D pose. Similar to GDR-Net [43], we parameterize the 3D rotation using its allocentric 6D representation R_{6d} [17, 43], and the 3D translation as the projected 3D centroid and the object's distance[20]. As backbone we leverage ResNet34 [6] for all experiments on the LM dataset [7], while we employ ResNeSt50 [48] for the more challenging datasets, *i.e.* LMO [3] and YCB-V [45].

After feature extraction using the aforementioned backbone, we append two decoders for estimation of self-occlusion and 2D-3D correspondences. The first branch essentially outputs 6-channel self-occlusion maps with a resolution of 64×64. The second branch predicts three different groups of intermediate geometric feature maps of size 64×64. While the first group describes the visible object mask, the other two groups describe the 2D-3D correspondence field [5] and surface region attention map as defined in [43]. Finally, self-occlusion and point matching feature maps are fed into the pose regression network to predict 6D pose directly. We adopt the identical pose regression network as in [43].

4.2. Training Details

Implementation Details. Our network is trained end-to-end using Ranger optimizer [22, 49, 46] on a single TITAN X GPU. We use a batch size of 24 and a base learning rate of 1e-4. We anneal the learning rate with a cosine schedule at 72% of the training phase. Unless specified otherwise, we set $\{\lambda_1, \lambda_2, \lambda_3\} = \{1/f, 10, 1\}$, where f denotes the focal length. Moreover, to increase stability, we first train the network without $\mathcal{L}_{cl-3D}$ and $\mathcal{L}_{cl-2D}$ and add them after 20% of the total training epochs. During training, color augmentation and mask erosion are randomly applied to avoid overfitting similar to [38]. For 2D localization we utilize Faster-RCNN [33] on LMO and FCOS [41] on YCB-V. Notice that we do not take special care of symmetric objects [43, 9] or post-refinement [19, 18].

Datasets. We test SO-Pose on three commonly-used datasets, *i.e.* LM [8], LMO [2] and YCB-V [45]. LM consists of individual sequences for 13 objects undergoing mild occlusion. We follow [43, 20] and employ $\approx 15\%$ of the RGB images for training and the remaining part for testing. We additionally render 1K synthetic images for each object during training. LMO extends LM by annotating one sequence with other 8 visible objects, often imposing severe occlusion on the objects. Similarly, we render additional 10k synthetic images for each object. Finally, YCB-V is a very challenging dataset exhibiting strong occlusion, clutter and several symmetric objects. We adopt the provided real images of 21 objects and publicly available physically-based rendered (*pbr*) data for training and testing as in [43].

We additionally evaluate our method following the BOP setup on LMO and YCB-V [11].

Evaluation Metrics. We employ the most commonly used metrics for comparison with other state-of-the-art methods. Thereby, *ADD(-S)* [8, 10] measures the percentage of transformed model points whose deviation from ground truth lies below 10% of the object's diameter (0.1d). For symmetric objects, *ADD(-S)* measures the deviation to the closet model point [8, 10]. Further, $n°n\,cm$ [35] measures the percentage of predicted 6D poses whose rotation error is less than $n°$ and translation error is below $n\,cm$. On YCB-V dataset, we also compute the AUC (area under curve) of ADD-S and ADD(-S) similar to [45]. For BOP settings on LMO and YCB-V, we additionally compute AR_{VSD}, AR_{MSSD}, AR_{MSPD} as proposed by [11]. We also provide the average AR score to compare the performance on various datasets.

4.3. Comparison with State of the Art

This section compares SO-Pose with other state-of-the-art methods on different datasets.

Results on LM. As shown in Tab. 1, our method consistently outperforms all baseline methods for each metric, especially in terms of $ADD(-S)\,0.02d$ and $2°2\,cm$. Compared to GDR-Net [43], under $ADD(-S)\,0.02d$, we improves from 35.3 to 45.9, up to 30%. Since these two strict metrics are usually utilized to measure performance in robotic grasping or other high-level tasks, the significant improvement of SO-Pose in Tab. 1 demonstrates that our method has great potential in robotic applications.

Results on LMO. We compare our method with state-of-the-art competitors in terms of $ADD(-S)$ in Tab. 3. When trained with *real+syn*, our method achieves even comparable results to refinement-based method such as DeepIM [19] and outperforms all other competitors. Further, using *real+pbr* data for training, our method achieves state-of-the-art performance on 5 out of 8 objects. Thereby, our average score surpasses all other methods by a large margin, with 62.3 against $24.9 - 56.1$.

Results on YCB-V. As for YCB-V, we show our results in Tab. 2. Using ResNeSt50 [48], we outperform again all other methods under ADD(-S) and AUC of ADD-S, with 56.8 and 90.9 against second best results 53.9 and 89.8. Under AUC of ADD(-S), we are only a little inferior to CosyPose [18], with 83.9 compared to 84.5. Nevertheless, whilst achieving comparable results as Cosy-Pose, out method runs significantly faster as CosyPose is a refinement-driven method, while we only need a single forward pass to obtain the final 6D pose.

Results under BOP metrics. In Tab. 4, we report our results under the BOP setup. To ensure a fair comparison with the related works, on LMO we only utilize the provided pbr data for training, whereas on YCB-V, both real and pbr data are utilized [9, 20, 43]. For all non-refinement

Method	ADD(-S)			$2°2cm$	$5°5cm$
	0.02d	0.05d	0.1d		
CDPN [20]	-	-	89.9	-	94.3
GDR-Net [43]	35.3	76.3	93.7	62.1	95.6
Ours(S)	36.6	76.8	94.0	59.1	97.0
Ours (w/o $\mathcal{L}_{cl-3D}$)	41.6	81.7	95.7	67.4	97.1
Ours (w/o $\mathcal{L}_{cl-2D}$)	44.7	81.3	95.5	73.1	98.0
Ours	**45.9**	**83.1**	**96.0**	**76.9**	**98.5**

Table 1. **Ablation Study on LM.** We provide results of our method with different loss terms. Ours(S) stands for removing both the $\mathcal{L}_{cl-3D}$ and $\mathcal{L}_{cl-2D}$ terms.

Method	P.E.	Ref.	ADD (-S)	AUC of ADD-S	AUC of ADD(-S)
PoseCNN [45]	1		21.3	75.9	61.3
SegDriven [14]	1		39.0	-	-
PVNet [28]	M		-	-	73.4
S.Stage [12]	M		53.9	-	-
GDR-Net [43]	1		49.1	89.1	80.2
DeepIM [19]	1	✓	-	88.1	81.9
CosyPose [18]	1	✓	-	89.8	**84.5**
Ours(34)	1		54.6	89.7	82.3
Ours(50)	1		**56.8**	**90.9**	83.9

Table 2. **Results on YCB-V.** We report the results of our method with different backbones. Ours(34) uses ResNet34 [6] while Ours(50) uses ResNeSt50 [48]. Ref. stands for refinement. P.E. reflects the training strategy of pose estimator, 1 represents single model for all objects while M represents one model per object. In general, the latter strategy benefits accuracy yet limits practical use.

methods, SO-Pose again achieves superior results reporting a mean AR of 0.664 compared to CDPN-v2 [20] with 0.578 and EPOS [9] with 0.621. Nevertheless, we are a little inferior to CosyPose, the overall best performing method that adopts sophisticated refinement.

4.4. Ablation Study

We conduct several ablations on each dataset. Thereby, except for ablated terms, we leave all other terms unchanged using the values from the experimental setup.

Effectiveness of cross-layer consistency. In Tab. 1 we demonstrate the effectiveness of the proposed cross-layer consistency terms on LM. We gradually remove L_{cl-3D} and L_{cl-2D} to observe their impact on the 6D pose w.r.t ADD(-S), $2°2\,cm$ and $5°5\,cm$. Thereby, we can observe that the accuracy decreases when removing either loss term, verifying the usefulness of our cross-layer consistencies.

Benefits of employing self-occlusion. To show that self-occlusion consistently improves pose quality, we additionally adopted the two-stage method CDPN [20] to also incorporate our two-layer representation. As CDPN is grounded on RANSAC/PnP to extract the 3D rotation from 2D-3D

Method	w/o Refinement								w/ Refinement	
	PoseCNN [45]	PVNet [28]	S.Stage [12]	HybridPose [36]	GDR-Net [43]		Ours (SO-Pose)		DPOD [47]	DeepIM [19]
P.E.	1	M	M	M	1	1	1	1	1	1
Training Data	real +syn	real +syn	real +syn	real +syn	real +syn	real +pbr	real +syn	real +pbr	real +syn	real +syn
Ape	9.6	15.8	19.2	20.9	41.3	44.9	46.3	48.4	-	**59.2**
Can	45.2	63.3	65.1	75.3	71.1	79.7	81.1	**85.8**	-	63.5
Cat	0.9	16.7	18.9	24.9	18.2	30.6	18.7	**32.7**	-	26.2
Driller	41.4	65.7	69.0	70.2	54.6	67.8	71.3	**77.4**	-	55.6
Duck	19.6	25.2	25.3	27.9	41.7	40.0	43.9	48.9	-	**52.4**
*Eggbox**	22.0	50.2	52.0	52.4	40.2	49.8	46.6	52.4	-	**63.0**
*Glue**	38.5	49.6	51.4	53.8	59.5	73.7	63.3	**78.3**	-	71.7
Holep.	22.1	36.1	45.6	54.2	52.6	62.7	62.9	**75.3**	-	52.5
Mean	24.9	40.8	43.3	47.5	47.4	56.1	54.3	**62.3**	47.3	55.5

Table 3. **Comparison with state-of-the-art methods on LMO.** We list the Average Recall of ADD(-S). (∗) denotes symmetric objects.

Method	P.E.	Ref.	LMO			YCB-V			Mean
			AR_{VSD}	AR_{MSSD}	AR_{MSPD}	AR_{VSD}	AR_{MSSD}	AR_{MSPD}	*AR*
CosyPose [18]	1	✓	**0.480**	<u>0.606</u>	0.812	**0.772**	**0.842**	**0.850**	**0.727**
EPOS [9]	1		0.389	0.501	0.750	0.626	0.677	<u>0.783</u>	0.621
PVNet [28]	M		0.428	0.543	0.754	-	-	-	-
CDPN-v2 [20]	M		<u>0.445</u>	**0.612**	<u>0.815</u>	0.396	0.570	0.631	0.578
GDR-Net [43]	1		-	-	-	0.584	0.674	0.726	-
Ours	1		0.442	0.581	**0.817**	<u>0.652</u>	<u>0.731</u>	0.763	<u>0.664</u>

Table 4. **Comparison with state-of-the-art methods on LMO and YCB-V under BOP metrics.** We provide results for AR_{VSD}, AR_{MSSD} and AR_{MSPD} on LMO and YCB-V. Mean AR represents the overall performance on these two datasets as the average over all AR scores. Overall best results are in bold and the second best results are underlined.

Method	ADD(-S)	$2°2cm$	$5°5cm$
CDPN*	92.82	68.56	97.03
CDPN*-ours (w/o $\mathcal{L}_{cdpn}$)	93.09	69.37	97.06
CDPN*-ours (w/ $\mathcal{L}_{cdpn}$)	**94.77**	**71.33**	**97.07**

Table 5. **Evaluation of our two-layer model o top of another baseline method CDPN [20].** We update the original CDPN to CDPN*. As for SO-Pose, we integrate our self-occlusion branch into the CDPN structure. Since CDPN predicts rotation with RANSAC/PnP, we rearrange Eq. 8 and Eq. 9 to derive a new loss term L_{cdpn}, as defined in Eq. 18.

correspondences, we slightly adjust our cross-layer consistency terms as follows,

$$\mathcal{L}_{cdpn} = \mathcal{L}_{cl-3D}(R \to \hat{R}) + \mathcal{L}_{cl-2D}(R \to \hat{R}) \quad (18)$$

Essentially, the consistency term $\mathcal{L}_{cdpn}$ is computed by replacing the predicted rotation R in $\mathcal{L}_{cl-3D}$ and $\mathcal{L}_{cl-2D}$ with the actual ground truth rotation $\hat{R}$. Except for $\mathcal{L}_{cdpn}$, all original loss terms in CDPN are preserved. As shown in Tab. 5, after introducing our two-layer model into CDPN, the performance improves again significantly for all metrics. This clearly demonstrates the generalizability of our proposed two-layer model.

Impact of different backbones. We report the results of our method with ResNet34 [6] and ResNeSt50 [48] as backbone in Tab. 2. Although the performance degrades slightly after changing ResNeSt50 to ResNet34, our method still outperforms most state-of-the-art methods, proving its efficacy regardless of the employed backbones.

4.5. Runtime Analysis

Given a 640×480 image from YCB-V with multiple objects, our method takes about 30ms to handle a single object and 50ms to process all objects in the image on an Intel 3.30GHz CPU and a TITAN X (12G) GPU. This includes the additional 15ms for 2D localization using Yolov3 [32]. As shown in Fig. 5, we demonstrate the Speed-AR score figure on YCB-V. Our method achieves second best results (AR: 0.715) in real time, which further verifies the great potential of our method in practical use.

4.6. Qualitative evaluation

We provide qualitative results for LMO in Fig. 4 and YCB-V in Fig. 6. In particular, in Fig. 4, we show four exemplary results for pose estimation together with the corresponding error maps for rendered 2D-3D correspondences with the estimated pose. In Fig. 4 (a) ape, due to wrong detection, the predicted 6D pose diverges from the ground
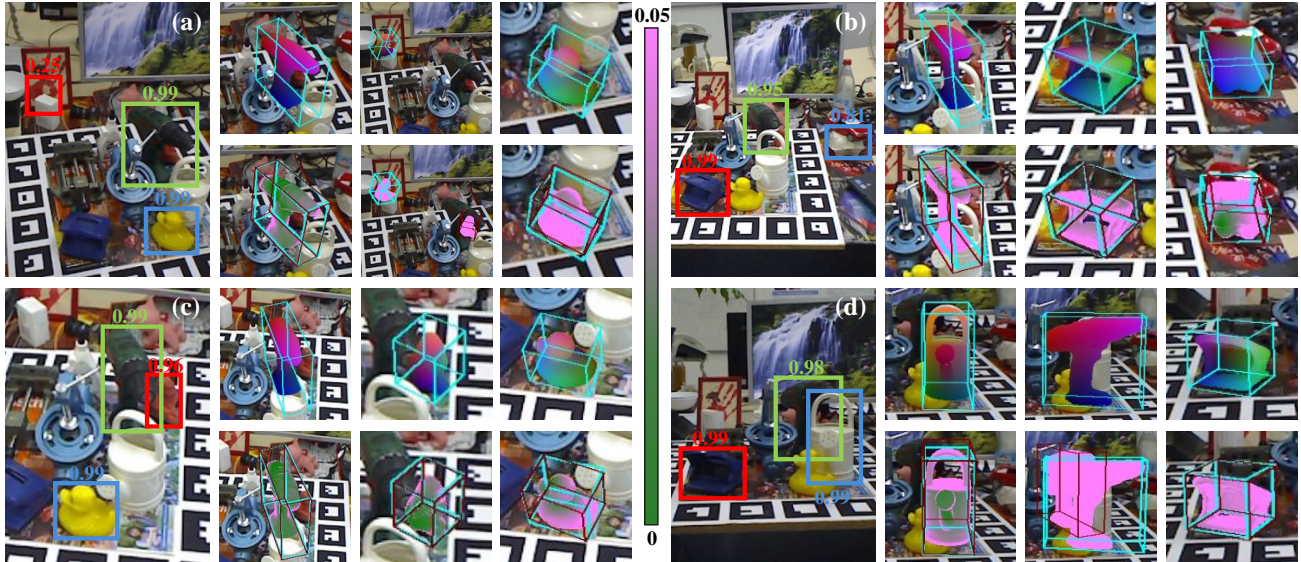
Figure 4. **Qualitative results on LMO.** We presents 4 examples of 6D pose estimation results. Given an input image, we first show the 2D detection results with confidence scores on top of the bounding boxes. Next to it we show the predicted 2D-3D matching (rendered with the predicted pose, color-coded) and object bounding box (estimated with the predicted pose, light blue) are shown on the top row. Correspondingly, we demonstrate the error of the predicted 2D-3D matching in the second row, from green to pink as illustrated in the middle line. We also visualize the ground truth bounding boxes (red).
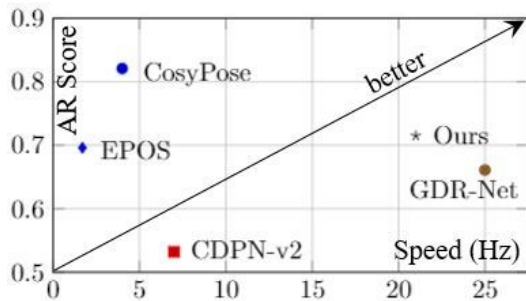
Figure 5. **Comparison of running speed (Hz) and AR score on YCB-V dataset**. We compare our method with CosyPose [18], GDR-Net [43], CDPN-v2 [20] and EPOS [9]. Along the direction of the arrow, method performs better, achieving higher accuracy in less inference time.
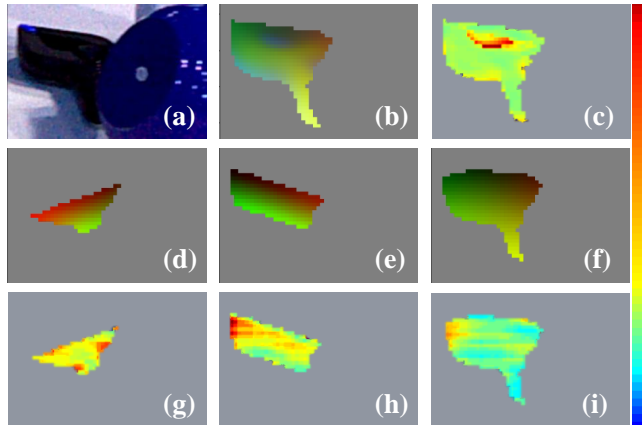

Figure 6. **Demonstration of the predicted two-layer model.** For object (a), we demonstrate its 2D-3D point matching in (b) and self-occlusion coordinates in (d), (e) and (f). (c), (g), (h), (i) are corresponding error maps of (b), (d), (e), (f). The color bar on the right side indicates the color-coded error for error maps. We normalize the error maps for better visualization, thus from the bottom to top of the color bar, the error ranges from 0 to 1.

truth completely. In (d) driller, we demonstrate wrong prediction of 6D pose due to strong occlusion. Finally in Fig. 6, we show the normalized two-layer representation of an object from YCB-V as predicted by our model. For more qualitative results, please refer to the Supplementary Material.

5. Conclusion

In this paper, we present a novel two-layer model that combines 2D-3D point correspondences and self-occlusion information to encapsulate explicitly the spatial cues of 3D object. Then based on the two-layer model, we establish SO-Pose, an end-to-end 6D pose regression framework that achieves significant improvements on various challeng-

ing datasets over other single-layer competitors. The experimental evaluation also demonstrates that our two-layer model is applicable to a wide range of 6D pose estimation frameworks and can consistently benefit the performance.

In the future, we plan to focus on integrating the two-layer model into self-supervised 6D pose estimation and category-level unseen object analysis.

References

[1] Pedram Azad, Tamim Asfour, and Ruediger Dillmann. Stereo-based 6d object localization for grasping with humanoid robot systems. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 919–924. IEEE, 2007. 1

[2] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D object pose estimation using 3D object coordinates. In *ECCV*, pages 536–551, 2014. 5

[3] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In *CVPR*, pages 3364–3372, 2016. 5

[4] Bo Chen, Alvaro Parra, Jiewei Cao, Nan Li, and Tat-Jun Chin. End-to-end learnable geometric vision by backpropagating pnp optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8100–8109, 2020. 1

[5] Alexander Grabner, Peter M. Roth, and Vincent Lepetit. 3D Pose Estimation and 3D Model Retrieval for Objects in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 5

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5, 6, 7

[7] 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. 5

[8] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer, 2012. 5, 6

[9] Tomas Hodan, Daniel Barath, and Jiri Matas. Epos: Estimating 6d pose of objects with symmetries. In *CVPR*, pages 11703–11712, 2020. 1, 2, 3, 5, 6, 7, 8

[10] Tomáš Hodaň, Jiří Matas, and Štěpán Obdržálek. On evaluation of 6d object pose estimation. *ECCVW*, pages 606–619, 2016. 6

[11] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018. 3, 6

[12] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2930–2939, 2020. 1, 2, 3, 6, 7

[13] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. In *CVPR*, 2019. 3

[14] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. In *CVPR*, pages 3385–3394, 2019. 6

[15] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017. 1, 2

[16] Wadim Kehl, Fausto Milletari, Federico Tombari, Slobodan Ilic, and Nassir Navab. Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In *ECCV*, 2016. 2

[17] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *CVPR*, 2018. 5

[18] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020. 1, 2, 5, 6, 7, 8

[19] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep iterative matching for 6d pose estimation. *IJCV*, pages 1–22, 2019. 2, 5, 6, 7

[20] Zhigang Li, Gu Wang, and Xiangyang Ji. CDPN: Coordinates-Based Disentangled Pose Network for Real-Time RGB-Based 6-DoF Object Pose Estimation. In *ICCV*, pages 7678–7687, 2019. 1, 2, 5, 6, 7, 8

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 2

[22] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*, 2019. 5

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 2

[24] Fabian Manhardt, Diego Martin Arroyo, Christian Rupprecht, Benjamin Busam, Tolga Birdal, Nassir Navab, and Federico Tombari. Explaining the ambiguity of object detection and 6d pose from visual data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6841–6850, 2019. 2

[25] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. ROI-10D: Monocular lifting of 2d detection to 6d pose and metric shape. In *CVPR*, pages 2069–2078, 2019. 1

[26] Fabian Manhardt, Wadim Kehl, Nassir Navab, and Federico Tombari. Deep model-based 6d pose refinement in rgb. In *ECCV*, 2018. 2

[27] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *ICCV*, 2019. 2

[28] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019. 2, 6, 7

[29] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-

d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 1

[30] Mahdi Rad and Vincent Lepetit. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. In *ICCV*, pages 3828–3836, 2017. 2

[31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 3

[32] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 3, 7

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 5

[34] Daeyun Shin, Zhile Ren, Erik B Sudderth, and Charless C Fowlkes. 3d scene reconstruction with multi-layer depth and epipolar transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2172–2182, 2019. 2, 3

[35] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, pages 2930–2937, 2013. 6

[36] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 431–440, 2020. 2, 7

[37] Martin Sundermeyer, Maximilian Durner, En Yen Puang, Zoltan-Csaba Marton, Narunas Vaskevicius, Kai O Arras, and Rudolph Triebel. Multi-path learning for object pose estimation across domains. In *CVPR*, pages 13916–13925, 2020. 2

[38] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018. 2, 5

[39] David Joseph Tan, Federico Tombari, and Nassir Navab. Real-time accurate 3d head tracking and pose estimation with consumer rgb-d cameras. *IJCV*, 126(2-4):158–183, 2018. 1

[40] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *CVPR*, pages 292–301, 2018. 2

[41] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019. 5

[42] Gu Wang, Fabian Manhardt, Jianzhun Shao, Xiangyang Ji, Nassir Navab, and Federico Tombari. Self6d: Self-supervised monocular 6d object pose estimation. In *ECCV*, Aug. 2020. 1

[43] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *CVPR*, June 2021. 1, 2, 3, 5, 6, 7, 8

[44] Paul Wohlhart and Vincent Lepetit. Learning Descriptors for Object Recognition and 3D Pose Estimation. In *CVPR*, 2015. 2

[45] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. *RSS*, 2018. 2, 5, 6, 7

[46] Hongwei Yong, Jianqiang Huang, Xiansheng Hua, and Lei Zhang. Gradient centralization: A new optimization technique for deep neural networks. In *European Conference on Computer Vision*, pages 635–652. Springer, 2020. 5

[47] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: Dense 6d pose object detector in rgb images. In *ICCV*, 2019. 1, 2, 7

[48] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Muller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020. 5, 6, 7

[49] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 5