GDR-Net: Geometry-Guided Direct Regression Network for Monocular 6D Object Pose Estimation

Gu Wang[1,2], Fabian Manhardt[2], Federico Tombari[2,3], Xiangyang Ji[1]
[1] Tsinghua University, BNRist [2] Technical University of Munich [3] Google
wangg16@mails.tsinghua.edu.cn, fabian.manhardt@tum.de,
tombari@in.tum.de, xyji@tsinghua.edu.cn

Abstract

6D pose estimation from a single RGB image is a fundamental task in computer vision. The current top-performing deep learning-based methods rely on an indirect strategy, i.e.*, first establishing 2D-3D correspondences between the coordinates in the image plane and object coordinate system, and then applying a variant of the PnP/RANSAC algorithm. However, this two-stage pipeline is not end-to-end trainable, thus is hard to be employed for many tasks requiring differentiable poses. On the other hand, methods based on direct regression are currently inferior to geometry-based methods. In this work, we perform an in-depth investigation on both direct and indirect methods, and propose a simple yet effective Geometry-guided Direct Regression Network (GDR-Net) to learn the 6D pose in an end-to-end manner from dense correspondence-based intermediate geometric representations. Extensive experiments show that our approach remarkably outperforms state-of-the-art methods on LM, LM-O and YCB-V datasets. The code will be available at https://git.io/GDR-Net.*

1. Introduction

Estimating the 6D pose, *i.e.* the 3D rotation and 3D translation, of objects with respect to the camera is a fundamental problem in computer vision. It has wide applicability to many real-world tasks such as robotic manipulation [10, 66, 53], augmented reality [35, 50] and autonomous driving [32, 58]. Most traditional methods rely on depth data for this task [13, 36, 14, 63, 54], while monocular methods lagged considerably behind [10, 12]. Nonetheless, with the advent of deep learning and especially the rise of Convolutional Neural Networks (CNNs), accuracy and robustness of monocular 6D object pose estimation have been consistently improving, even at times surpassing methods relying on depth data [22, 42, 18].

Different strategies for predicting 6D pose from monocular data have been proposed. For instance, learning of an embedding space for pose [49] or direct regression of the 3D rotation and translation [31]. While these methods generally perform well, they usually lack in accuracy when compared with approaches that instead rely on establishing 2D-3D correspondences prior to estimating the 6D pose [28, 15].

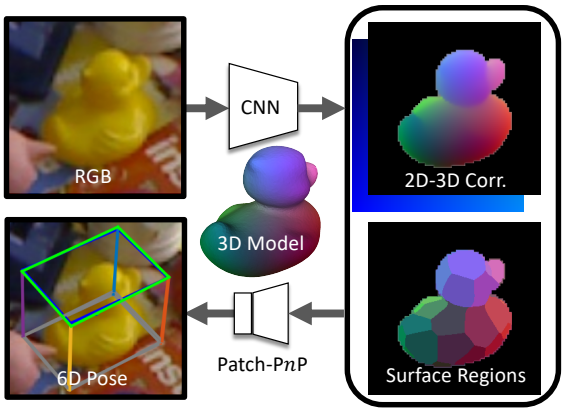


Figure 1: **Illustration of GDR-Net.** We directly regress the 6D object pose from a single RGB using a CNN and the learnable Patch-PnP by leveraging the guidance of intermediate geometric features including 2D-3D dense correspondences and surface region attention.

Differently, this latter class of methods usually involves solving the 6D pose through a variant of the PnP/RANSAC algorithm. While such a paradigm provides good estimates, it also suffers from several drawbacks. First, these methods are usually trained with a surrogate objective for correspondence regression, which does not necessarily reflect the actual 6D pose error after optimization. In practice, two sets of correspondences can have the same average error while describing completely different poses. Second, these approaches are not differentiable with respect to the estimated

6D pose, which limits learning. For instance, these methods cannot be coupled with self-supervised learning from unlabeled real data [55, 34, 1], as they require the computation of the pose to be fully differentiable in order to obtain a signal between data and pose. Finally, the RANSAC iterative process can be very time-consuming when dealing with dense correspondences.

To summarize, while methods grounded on 2D-3D correspondences are dominating the field, they still exhibit downsides due to the decoupling of the problem into two separate steps one of which not differentiable. Consequently, some efforts have been devoted to enabling back-propagation through the PnP/RANSAC stage. However, this either requires a complex training strategy in order to have good initialization of scene coordinates [4, 6, 7], or can only handle sparse correspondences of a predefined set of keypoints [8]. Recently, the authors of [19] proposed to leverage PointNet [41] in order to approximate PnP for sparse correspondences. While this proves to work well, PointNet disregards the fact that the correspondences are organized with respect to the image pixels, which tends to strongly deteriorate the performance as shown in [30].

In this work, we propose to overcome these limitations by establishing 2D-3D correspondences whilst computing the final 6D pose estimate in a fully differentiable way (Fig. 1). In its core, we propose to learn the PnP optimization, exploiting the fact that the correspondences are organized in image space, which gives a significant boost in performance, outperforming all prior works. To summarize, we make the following contributions:

- We revisit the key ingredients in direct 6D pose regression and observe that by choosing appropriate representations for the pose parameters, methods based on direct regression show competitive performance compared with state-of-the-art correspondence-based indirect methods.

- We further propose a simple yet effective Geometry-guided Direct Regression Network (GDR-Net) to boost the performance of direct 6D pose regression via leveraging the geometric guidance from dense correspondence-based intermediate representations.

Extensive experiments on LM [13], LM-O [3], and YCB-V [60] datasets show that our unified GDR-Net approach can achieve accurate, yet real-time and robust, monocular 6D object pose estimation.

2. Related Work

While methods based on depth data used to dominate the field of 6D pose estimation [13, 36, 14, 63, 54], Deep Learning-based methods have recently demonstrated promising results for the task at hand [17, 18]. In this section, we review several commonly employed strategies for monocular 6D pose estimation.

Indirect Methods. The most popular approach is to establish 2D-3D correspondences, which are then leveraged to solve for the 6D pose using a variant of the RANSAC-based PnP algorithm. For instance, [42] and [51] compute the 2D projections of a set of fixed control points (*e.g.* the 3D corners of the encapsulating bounding box). To enhance the robustness, [20] and [40] additionally conduct segmentation coupled with voting for each correspondence. However, the recent trend goes towards predicting dense rather than sparse correspondences [62, 28]. Moreover, while [38] leverages a GAN on top of dense correspondences to increase stability, [15] makes use of fragments in order to account for ambiguities in pose.

Another orthogonal line of works aims at learning a latent embedding of pose which can be utilized for retrieval during inference. These embeddings are commonly either grounded on metric learning employing a triplet loss [57], or via training of an Auto-Encoder [49, 48].

Direct Methods. Although indirect methods, leveraging 2D-3D correspondences, are currently performing better, they cannot be directly employed in many tasks, which require the pose estimation to be differentiable [55]. Hence, some methods directly regress the 6D pose, either leveraging a point matching loss [60, 27, 25] or employing separate loss terms for each component [31, 11]. Other methods discretize the pose space and conduct classification rather than regression [22]. A few methods also try to solve a proxy task during optimization. Thereby, [33] proposes to employ an edge-alignment loss using the distance transform, while [55] harnesses differentiable rendering to allow training on unlabeled samples.

Differentiable Indirect Methods. Recently, a few works attempt to make PnP/RANSAC differentiable. In [4, 6, 7] the authors introduce a novel differentiable way to apply RANSAC via sharing of hypotheses based on the predicted distribution. Nonetheless, these approaches require a complex training strategy, as they expect a good initialization for the scene coordinates. As for PnP, [8] employs the Implicit Function Theorem [23] to enable the computation of analytical gradients w.r.t. the pose loss. Yet, it is computationally expensive especially given too many correspondences since PnP/RANSAC is still needed for both training and inference. Instead, [19] attempts to learn the PnP stage with a PointNet-based architecture [41] which learns to infer the 6D pose from a fixed set of sparse 2D-3D correspondences.

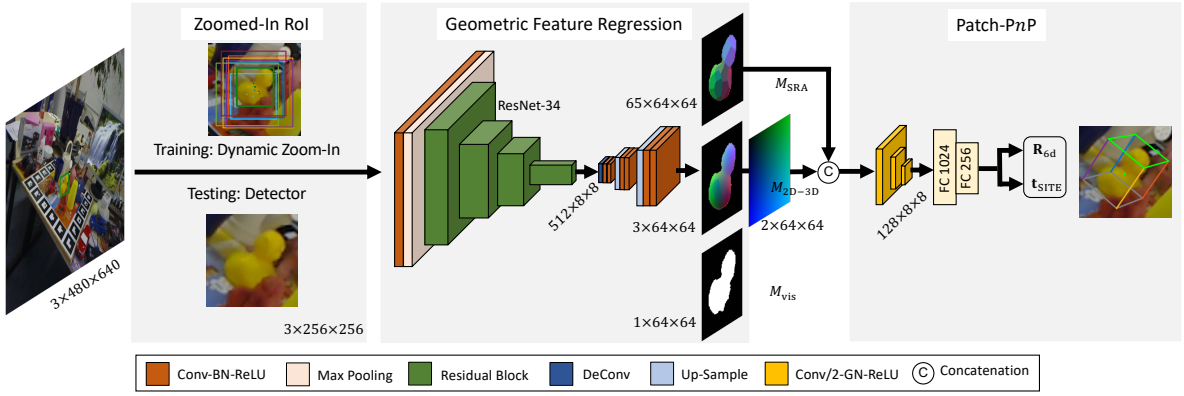


Figure 2: **Framework of GDR-Net.** Given an RGB image I, our GDR-Net takes the zoomed-in RoI (Dynamic Zoom-In for training, off-the-shelf detections for testing) as input and predicts several intermediate geometric features. Then the Patch-PnP directly regresses the 6D object pose from *Dense Correspondences* ($M_{2D\text{-}3D}$) and *Surface Region Attention* (M_{SRA}).

Beyond Instance-level 6D Pose Estimation. Noteworthy, a few methods are recently trying to go beyond the instance-level scenario, even estimating the pose [56], sometimes paired with shape [9, 34, 37], for previously unseen objects.

3. Method

Given an RGB image I and a set of N objects $\mathcal{O} = \{\mathcal{O}_i \mid i = 1, \cdots, N\}$ together with their corresponding 3D CAD models $\mathcal{M} = \{\mathcal{M}_i \mid i = 1, \cdots, N\}$, our goal is to estimate the 6D object pose $\mathbf{P} = [\mathbf{R}|\mathbf{t}]$ w.r.t. the camera for each object $\mathcal{O}$ present in I. Notice that $\mathbf{R}$ describes the 3D rotation and $\mathbf{t}$ denotes the 3D translation of the detected object.

Fig. 2 presents a schematic overview of the proposed methodology. In the core, we first detect all objects of interest using an off-the-shelf object detector, such as [44, 52]. For each detection, we then zoom in to the corresponding Region of Interest (RoI) and feed it to our network to predict several intermediate geometric feature maps. Finally, we directly regress the associated 6D object pose from the dense correspondence-based intermediate geometric features.

In the following, we first (Sec. 3.1) revisit the key ingredients of direct 6D object pose estimation methods. Afterwards (Sec. 3.2), we illustrate a simple yet effective Geometry-Guided Direct Regression Network (GDR-Net) which unifies regression-based *direct* methods and geometry-based *indirect* methods, thus harnessing the best of both worlds.

3.1. Revisiting Direct 6D Object Pose Estimation

Direct 6D pose estimation methods usually differ in one or more of the following components. Firstly, the parameterization of the rotation $\mathbf{R}$ and translation $\mathbf{t}$, and secondly, the employed loss for pose. In this section, we investigate different commonly used parameterizations and demonstrate that appropriate choices have significant impact on the 6D pose estimates.

Parameterization of 3D Rotation. Several different parameterization can be employed to describe 3D rotations. Since many representations exhibit ambiguities, *i.e.* $\mathbf{R}_i$ and $\mathbf{R}_j$ describe the same rotation with $\mathbf{R}_i \neq \mathbf{R}_j$, most works rely on parametrizations that are unique to help training. Therefore, common choices are unit quaternions [60, 33, 27], log quaternions [37], or Lie algebra-based vectors [11].

Nevertheless, it is well-known that all representations with four or fewer dimensions for 3D rotation have discontinuities in the Euclidean space. When regressing a rotation, this introduces an error close to the discontinuities which becomes often significantly large. To overcome this limitation, [65] proposed a novel continuous 6-dimensional representation for $\mathbf{R}$ in $SO(3)$, which has proven promising [65, 25]. Specifically, the 6-dimensional representation $\mathbf{R}_{6d}$ is defined as the first two columns of $\mathbf{R}$

$$\mathbf{R}_{6d} = [\mathbf{R}_{.1} \mid \mathbf{R}_{.2}] \,. \tag{1}$$

Given a 6-dimensional vector $\mathbf{R}_{6d} = [\mathbf{r}_1|\mathbf{r}_2]$, the rotation matrix $\mathbf{R} = [\mathbf{R}_{.1}|\mathbf{R}_{.2}|\mathbf{R}_{.3}]$ can be computed according to

$$\begin{cases} \mathbf{R}_{.1} = \phi(\mathbf{r}_1) \\ \mathbf{R}_{.3} = \phi(\mathbf{R}_{.1} \times \mathbf{r}_2) \\ \mathbf{R}_{.2} = \mathbf{R}_{.3} \times \mathbf{R}_{.1} \end{cases} , \tag{2}$$

where $\phi(\bullet)$ denotes the vector normalization operation.

Given the advantages of this representation, in this work we employ $\mathbf{R}_{6d}$ to parameterize the 3D rotation. Nevertheless, in contrast to [65, 25], we propose to let the network

predict the allocentric representation [24] of rotation $\mathbf{R}_{\text{a6d}}$. This representation is favored as it is viewpoint-invariant under 3D translations of the object. Hence, it is more suitable to deal with zoomed-in RoIs. Note that the egocentric rotation can be easily converted from allocentric rotation given 3D translation and camera intrinsics K following [24].

Parameterization of 3D Translation. Since directly regressing the translation $\mathbf{t} = [t_x, t_y, t_z]^T \in \mathbb{R}^3$ in 3D space does not work well in practice, previous works usually decouple the translation into the 2D location (o_x, o_y) of the projected 3D centroid and the object's distance t_z towards the camera. Given the camera intrinsics K, the translation can be calculated via back-projection

$$\mathbf{t} = K^{-1} t_z \left[o_x, o_y, 1\right]^T . \tag{3}$$

Exemplary, [22, 49] approximate (o_x, o_y) as the bounding box center (c_x, c_y) and estimate t_z using a reference camera distance. PoseCNN [60] directly regresses (o_x, o_y) and t_z. Nonetheless, this is not suitable for dealing with zoomed-in RoIs, since it is essential for the network to estimate position and scale invariant parameters.

Therefore, in our work we utilize a Scale-Invariant representation for Translation Estimation (SITE) [28]. Concretely, given the size $s_o = \max(w, h)$ and center (c_x, c_y) of the detected bounding box and the ratio $r = s_{\text{zoom}}/s_o$ w.r.t. the zoom-in size s_{zoom}, the network regresses the scale-invariant translation parameters $\mathbf{t}_{\text{SITE}} = [\delta_x, \delta_y, \delta_z]^T$, where

$$\begin{cases} \delta_x = (o_x - c_x)/w \\ \delta_y = (o_y - c_y)/h \\ \delta_z = t_z/r \end{cases} . \tag{4}$$

Finally, the 3D translation can be solved according to Eq. 3.

Disentangled 6D Pose Loss. Apart from the parameterization of rotation and translation, the choice of loss function is also crucial for 6D pose optimization. Instead of directly utilizing distances based on rotation and translation (*e.g.*, angular distance, L_1 or L_2 distances), most works employ a variant of Point-Matching loss [27, 60, 25] based on the ADD(-S) metric [13, 3] in an effort to couple the estimation of rotation and translation.

Inspired by [46, 25], we employ a novel variant of disentangled 6D pose loss via individually supervising the rotation $\mathbf{R}$, the scale-invariant 2D object center (δ_x, δ_y), and the distance δ_z.

$$\mathcal{L}_{\text{Pose}} = \mathcal{L}_{\mathbf{R}} + \mathcal{L}_{\text{center}} + \mathcal{L}_z . \tag{5}$$

Thereby,

$$\begin{cases} \mathcal{L}_{\mathbf{R}} & = \underset{\mathbf{x} \in \mathcal{M}}{\text{avg}} \|\hat{\mathbf{R}}\mathbf{x} - \bar{\mathbf{R}}\mathbf{x}\|_1 \\ \mathcal{L}_{\text{center}} & = \|(\hat{\delta}_x - \bar{\delta}_x, \hat{\delta}_y - \bar{\delta}_y)\|_1 \\ \mathcal{L}_z & = \|\hat{\delta}_z - \bar{\delta}_z\|_1 \end{cases} , \tag{6}$$

where $\hat{\bullet}$ and $\bar{\bullet}$ denote prediction and ground truth, respectively. To account for symmetric objects, given $\bar{\mathcal{R}}$, the set of all possible ground-truth rotations under symmetry, we further extend our loss to a symmetry-aware formulation $\mathcal{L}_{\mathbf{R},\text{sym}} = \min_{\bar{\mathbf{R}} \in \bar{\mathcal{R}}} \mathcal{L}_{\mathbf{R}}(\hat{\mathbf{R}}, \bar{\mathbf{R}})$.

3.2. Geometry-guided Direct Regression Network

In this section, we present our Geometry-guided Direct Regression Network, which we dub GDR-Net. Harnessing dense correspondence-based geometric features, we directly regress 6D object pose. Thereby, GDR-Net unifies approaches based on dense correspondences and direct regression.

Network Architecture. As shown in Fig. 2, we feed the GDR-Net with a zoomed-in RoI of size 256×256 and predict three intermediate geometric feature maps with spatial size of 64×64, which are composed of the *Dense Correspondences Map* ($M_{\text{2D-3D}}$), the *Surface Region Attention Map* (M_{SRA}) and the *Visible Object Mask* (M_{vis}).

Our network is inspired by CDPN [28], a state-of-the-art dense correspondence-based method for indirect pose estimation. In essence, we keep the layers for regressing M_{XYZ} and M_{vis}, while removing the disentangled translation head. Additionally, we append the channels required by M_{SRA} to the output layer. Since these intermediate geometric feature maps are all organized 2D-3D correspondences w.r.t. the image, we employ a simple yet effective 2D convolutional *Patch-PnP* module to directly regress the 6D object pose from $M_{\text{2D-3D}}$ and M_{SRA}.

The Patch-PnP module consists of three convolutional layers with kernel size 3×3 and stride $= 2$, each followed by Group Normalization [59] and ReLU activation. Two Fully Connected (FC) layers are then applied to the flattened feature, reducing the dimension from 8192 to 256. Finally, two parallel FC layers output the 3D rotation $\mathbf{R}$ parameterized as $\mathbf{R}_{\text{6d}}$ (Eq. 1) and 3D translation $\mathbf{t}$ parameterized as $\mathbf{t}_{\text{SITE}}$ (Eq. 4), respectively.

Dense Correspondences Maps ($M_{\text{2D-3D}}$). In order to compute the Dense Correspondences Maps $M_{\text{2D-3D}}$, we first estimate the underlying Dense Coordinates Maps (M_{XYZ}). $M_{\text{2D-3D}}$ can then be derived by stacking M_{XYZ} onto the corresponding 2D pixel coordinates. In particular, given the CAD model of an object, M_{XYZ} can be obtained by rendering the model's 3D object coordinates given the associated

pose. Similar to [28, 56], we let the network predict a normalized representation of M_{XYZ}. Concretely, each channel of M_{XYZ} is normalized within $[0, 1]$ by (l_x, l_y, l_z), which is the size of corresponding tight 3D bounding box of the CAD model.

Notice that $M_{\text{2D-3D}}$ does not only encode the 2D-3D correspondences, but also explicitly reflect the geometric shape information of objects. Moreover, as previously mentioned, since $M_{\text{2D-3D}}$ is regular w.r.t. the image, we are capable of learning the 6D object pose via a simple 2D convolutional neural network (Patch-PnP).

Surface Region Attention Maps (M_{SRA}). Inspired by [15], we let the network predict the surface regions as additional ambiguity-aware supervision. However, instead of coupling them with RANSAC, we use them within our Patch-PnP framework.

Essentially, the ground-truth regions M_{SRA} can be derived from M_{XYZ} employing farthest points sampling.

For each pixel we classify the corresponding regions, thus the probabilities in the predicted M_{SRA} implicitly represent the symmetry of an object. For instance, if a pixel is assigned to two potential fragments due to a plane of symmetry, Minimizing this assignment will return a probability of 0.5 for each fragment. Moreover, leveraging M_{SRA} not only mitigates the influence of ambiguities but also acts as an auxiliary task on top of M_{3D}. In other words, it eases the learning of M_{3D} by first locating coarse regions and then regressing finer coordinates. We utilize M_{SRA} as a symmetry-aware attention to guide the learning of Patch-PnP.

Geometry-guided 6D Object Pose Regression. The presented image-based geometric feature patches, *i.e.*, $M_{\text{2D-3D}}$ and M_{SRA}, are then utilized to guide our proposed Patch-PnP for direct 6D object pose regression as

$$\mathbf{P} = \text{Patch-P}n\text{P}(M_{\text{2D-3D}}, M_{\text{SRA}}). \tag{7}$$

We employ L_1 loss for normalized M_{XYZ} and visible masks M_{vis}, and cross-entropy loss (CE) for M_{SRA}.

$$\begin{aligned}\mathcal{L}_{\text{Geom}} = \; & \|\bar{M}_{\text{vis}} \odot (\hat{M}_{\text{XYZ}} - \bar{M}_{\text{XYZ}})\|_1 + \|\hat{M}_{\text{vis}} - \bar{M}_{\text{vis}}\|_1 \\ & + CE(\bar{M}_{\text{vis}} \odot \hat{M}_{\text{SRA}}, \bar{M}_{\text{SRA}}).\end{aligned} \tag{8}$$

Thereby, $\odot$ denotes element-wise multiplication and we only supervise M_{XYZ} and M_{SRA} using the visible region.

The overall loss for GDR-Net can be summarized as $\mathcal{L}_{\text{GDR}} = \mathcal{L}_{\text{Pose}} + \mathcal{L}_{\text{Geom}}$. Notice that our GDR-Net can be trained end-to-end, without requiring any three-stage training strategy as in [28].

Decoupling Detection and 6D Object Pose Estimation. Similar to [28, 25], we mainly focus on the network for 6D object pose estimation and make use of an existing 2D object detector to obtain the zoomed-in input RoIs. This allows us to directly make use of the advances in runtime [43, 2] and accuracy [44, 52] within the rapidly growing field of 2D object detection, without having to change or re-train the pose network. Therefore, we adopt a simplified Dynamic Zoom-In (DZI) [28] to decouple the training of our GDR-Net and object detectors. During training, we first uniformly shift the center and scale of the ground-truth bounding boxes by a ratio of 25%. We then zoom in the input RoIs with a ratio of $r = 1.5$ while maintaing the original aspect ratio. This ensures that the area containing the object is approximately half the RoI. DZI can also circumvent the need of dealing with varying object sizes.

Noteworthy, although we employ a two-stage approach, one could also implement GDR-Net on top of any object detector and train it in an end-to-end manner.

4. Experiments

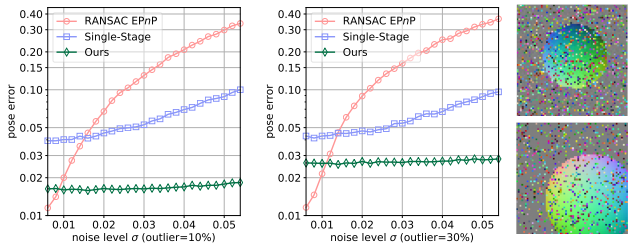
In this section we first introduce our experimental setup, and then present the evaluation results for several commonly employed benchmark datasets. Thereby, we first present experiments on a synthetic toy dataset, which clearly demonstrates the benefit of our Patch-PnP compared to the classic optimization-driven PnP. Additionally, we demonstrate the effectiveness of our individual components by performing an ablative study on LM [13]. Finally, we compare GDR-Net with state-of-the-art methods on two challenging datasets, *i.e.* LM-O [3] and YCB-V [60].

4.1. Experimental Setup

Implementation Details. All our experiments are implemented using PyTorch [39]. We train all our networks end-to-end using the Ranger optimizer [29, 64, 61][1] with a batch size of 24 and a base learning rate of 1e-4, which we anneal at 72% of the training phase using a cosine schedule [21].

Datasets. We conduct our experiments on four datasets: Synthetic Sphere [26, 19], LM [13], LM-O [3], and YCB-V [60]. *The Synthetic Sphere dataset* contains 20k samples for training and 2k for testing, created by randomly capturing a unit sphere model using a virtual calibrated camera with focal length 800, resolution 640×480, and the principal point located at the image center. The Rotations and translations are uniformly sampled in 3D space, and within an interval of $[-2, 2] \times [-2, 2] \times [4, 8]$, respectively. *LM dataset* consists of 13 sequences, each containing $\approx$ 1.2k images with ground-truth poses for a single object with clutter and mild occlusion. We follow [5] and employ ≈15% of the RGB images for training and 85% for testing. We additionally use 1k rendered RGB images for each object during

[1] Ranger means the RAdam [29] optimizer combined with Lookahead [64] and Gradient Centralization [61].



(a) (b) (c)

Figure 3: **Results of PnP variants on Synthetic Sphere.** *(a, b)*: We compare our Patch-PnP module with the traditional RANSAC EPnP [26] and another learning-based PnP [19]. The pose error is reported as relative ADD error w.r.t. the sphere's diameter (y-axis in log-scale). *(c)*: Zoomed-In (64×64) synthetic examples for Patch-PnP.

training as in [28]. *LM-O* consists of 1214 images from a LM sequence, where the ground-truth poses of 8 visible objects with more occlusion are provided for testing. *YCB-V* is a very challenging dataset exhibiting strong occlusion, clutter and several symmetric objects. It comprises over 110k real images captured with 21 objects, both with and without texture. For both LM-O and YCB-V, we also leverage the publicly available synthetic data using physically-based rendering [18] for training.

Evaluation Metrics. We use two common metrics for 6D object pose evaluation, *i.e.* ADD(-S) [13, 16], and $n°, n$ cm [45]. *The ADD metric* [13] measures whether the average deviation of the transformed model points is less than 10% of the object's diameter (0.1d). For symmetric objects, the *ADD-S* metric is employed to measure the error as the average distance to the closest model point [13, 16]. When evaluating on YCB-V, we also compute the *AUC (area under curve) of ADD(-S)* by varying the distance threshold with a maximum of 10 cm [60]. *The $n°, n$ cm metric* [45] measures whether the rotation error is less than $n°$ and the translation error is below n cm. Notice that to account for symmetries, $n°, n$ cm is computed w.r.t. the smallest error for all possible ground-truth poses [27].

4.2. Toy Experiment on Synthetic Sphere

We conduct a toy experiment comparing our approach with PnP/RANSAC and [19] on the Synthetic Sphere dataset. We generate M_{XYZ} from the provided poses and feed them to our Patch-PnP. For fairness, M_{SRA} is excluded from the input. Following [19], during training, we randomly add Gaussian noise $\mathcal{N}(0, \sigma^2)$ with $\sigma \in \mathcal{U}[0, 0.03]$ to each point of the dense coordinates maps. Since the coordinates maps are normalized in $[0, 1]$, we choose 0.03 as it reflects approximately the same level of noise as in [19]. Additionally, we randomly generated 0% to 30% of outliers for M_{XYZ} (Fig. 3c). During testing, we report the relative ADD error w.r.t. the sphere's diameter on the test set with different levels of noise and outliers.

Comparison with PnP/RANSAC and [19]. In Fig. 3, we demonstrate the effectiveness and robustness of our approach by comparing Patch-PnP with the traditional RANSAC-based EPnP [26] and the learning-based PnP from [19]). As depicted in Fig. 3, while RANSAC-based EPnP is more accurate when noise is unrealistically minimal, learning-based PnP methods are much more accurate and robust as the level of noise increases. Moreover, Patch-PnP is significantly more robust than Single-Stage [19] w.r.t. to noise and outliers, thanks to our geometrically rich and dense correspondences maps.

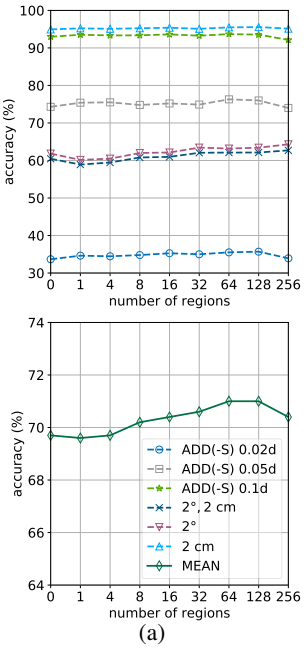
4.3. Ablation Study on LM

We present several ablation experiments for the widely used LM dataset [13]. We train a single GDR-Net for all objects for 160 epochs without applying any color augmentation. For fairness in evaluation, we leverage the detection results from Faster-RCNN as provided by [28].

Number of Regions in M_{SRA}. In Tab. 1a, we show results for different numbers of regions in M_{SRA}. Thereby, without our attention M_{SRA} (number of regions = 0), the accuracy is deliberately good, which suggests the effectiveness and versatility of Patch-PnP. Nevertheless, the overall accuracy can be further improved with increasing number of regions in M_{SRA}, despite starting to saturate around 64 regions. Thus, we use 64 regions for M_{SRA} in all other experiments as trade-off between accuracy and memory.

Effectiveness of Patch-PnP. We demonstrate the effectiveness of the image-like geometric features ($M_{\text{2D-3D}}, M_{\text{SRA}}$) by comparing our Patch-PnP with traditional PnP/RANSAC [28] and the PointNet-like [41] PnP from [19]. For the latter, we extend the PointNet in [19] to account for dense correspondences. Specifically, we utilize PointNet to pointwisely transform the spatially flattened geometric features ($M_{\text{2D-3D}}$ and M_{SRA}) and directly predict the 6D pose with global max pooling followed by two FC layers. Since the correspondences are explicitly encoded in $M_{\text{2D-3D}}$, no special attention is needed for the keypoint orders as in [19].

As shown in Tab. 1b, Patch-PnP is more accurate than traditional PnP/RANSAC (B0 *vs*. A0) as well as PointNet-like PnP (B0 *vs*. C0) in estimating the 6D pose. Furthermore, in terms of rotation, our Patch-PnP outperforms PointNet-like PnP by a large margin, which proves the importance of exploiting the ordering within the correspondences.



(a)

Row	Method	ADD(-S) 0.02d	ADD(-S) 0.05d	ADD(-S) 0.1d	2°, 2 cm	2°	2 cm	MEAN
A0	CDPN [28]	-	-	89.9	-	-	92.8	-
B0	GDR-Net (**Ours**)	35.5	76.3	93.7	62.1	63.2	95.5	**71.0**
B1	B0: $\rightarrow$ Test with PnP/RANSAC	31.0	72.1	92.2	67.1	68.9	94.5	**71.0**
B2	B0: $\rightarrow$ Patch-PnP for $\mathbf{t}$; PnP/RANSAC for $\mathbf{R}$	35.6	76.0	93.6	67.1	69.0	95.5	**72.8**
C0	B0: Patch-PnP $\rightarrow$ PointNet-like PnP	29.2	72.6	92.3	44.5	45.8	94.3	63.1
D0	B0: $\mathbf{R}_{\text{a6d}} \rightarrow$ Egocentric $\mathbf{R}_{\text{6d}}$	36.1	75.7	93.2	60.4	61.5	95.3	70.4
D1	B0: $\mathbf{R}_{\text{a6d}} \rightarrow$ Allocentric quaternion	24.8	67.4	90.5	35.5	36.9	92.2	57.9
D2	B0: $\mathbf{R}_{\text{a6d}} \rightarrow$ Allocentric log quaternion	22.7	64.6	88.9	33.7	35.4	90.9	56.0
D3	B0: $\mathbf{R}_{\text{a6d}} \rightarrow$ Allocentric Lie algebra vector	23.0	66.3	89.7	33.8	35.3	91.4	56.6
E0	B0: $\mathbf{t}_{\text{SITE}} \rightarrow \mathbf{t}$	28.3	72.0	92.4	61.6	63.2	94.6	68.7
E1	B0: $\mathbf{t}_{\text{SITE}} \rightarrow (o_x, o_y); t_z$	31.4	73.7	93.3	50.4	51.6	94.7	65.8
E2	B0: $\delta_z \rightarrow t_z$	32.8	73.5	93.3	63.3	64.8	94.9	70.4
F0	B0: $\mathcal{L}_{\text{Pose}} \rightarrow \mathcal{L}_{\text{PM}} = \underset{\mathbf{x}\in\mathcal{M}}{\text{avg}} \lVert(\hat{\mathbf{R}}\mathbf{x}+\hat{\mathbf{t}}) - (\bar{\mathbf{R}}\mathbf{x}+\bar{\mathbf{t}})\rVert_1$	33.7	76.5	94.1	47.4	48.2	95.8	65.9
F1	F0: $\mathcal{L}_{\text{PM}} \rightarrow$ Disentangling $\mathbf{R}$; $\mathbf{t}$	30.8	71.1	91.8	64.6	66.8	93.5	69.8
F2	F0: $\mathcal{L}_{\text{PM}} \rightarrow$ Disentangling $\mathbf{R}$; $(t_x, t_y); t_z$	32.2	73.9	93.6	63.8	65.3	94.8	70.6
F3	B0: $\mathcal{L}_{\mathbf{R}} \rightarrow$ Angular loss	32.4	75.5	93.8	40.2	40.9	95.7	63.1
F4	B0: $\mathcal{L}_{\mathbf{R}} \rightarrow \mathcal{L}_{\mathbf{R},\text{sym}}$	35.5	75.8	93.9	61.6	62.7	95.4	70.8
G0	B0: $\mathcal{L}_{\text{GDR}} \rightarrow$ w/o $\mathcal{L}_{\text{Geom}}$	30.8	72.7	92.2	45.9	46.8	94.1	63.7
G1	G0: $\rightarrow$ w/o M_{2D}	18.6	60.1	85.6	26.0	27.8	87.6	51.0
G2	G0: $\mathbf{R}_{\text{a6d}} \rightarrow$ Allocentric quaternion	6.7	40.6	73.2	6.2	7.4	75.6	34.9
H0	B0: Faster-RCNN [44] $\rightarrow$ Yolov3 [43]	33.9	75.6	93.7	60.9	62.1	95.2	70.2

(b)

Table 1: **Ablation Study on LM.** *(a):* Ablation of number of regions in M_{SRA}. *(b):* Ablation of PnP type, the parameterization of $\mathbf{R}$ and $\mathbf{t}$, loss type and geometric guidance.

Parameterization of 6D Pose. In Tab. 1b, we illustrate the impact of our proposed 6D pose parameterization. In particular, the 6-dimensional $\mathbf{R}_{\text{6d}}$ (Eq. 1) achieves a much more accurate estimate of $\mathbf{R}$ than commonly used representations such as unit quaternions [60, 27], log quaternions [37] and the Lie algebra-based vectors [11] (*c.f.* B0 *vs*. D1-D3, and G0 *vs*. G2). Moreover, we can deduce that the allocentric representation is significantly stronger than the egocentric formulation (B0 *vs*. D0).

Similarly, the parameterization of the 3D translation is of high importance. Essentially, directly predicting $\mathbf{t}$ in 3D space leads to worse results than leveraging the scale-invariant formulation $\mathbf{t}_{\text{SITE}}$ (E0 *vs*. B0). Additionally, replacing the scale-invariant δ_z in $\mathbf{t}_{\text{SITE}}$ with the absolute distance t_z or directly regressing the object center (o_x, o_z) leads to inferior poses w.r.t. translation (B0 *vs*. E1, E2). Hence, when dealing with zoomed-in RoIs, it is essential to parameterize the 3D translation in a scale-invariant fashion.

Ablation on Pose Loss. As mentioned in Section 3.1, the loss function has an impact on direct 6D pose regression. In Tab. 1b, we compare our disentangled $\mathcal{L}_{\text{Pose}}$ to a simple angular loss and the Point-Matching loss [27] (F0). Furthermore, we present its disentangled versions following [46]. As shown in (B0 and F0-F4), all variants of the PM loss are clearly better than the angular loss in terms of rotation estimation. In addition, disentangling the rotation $\mathbf{R}$ and distance t_z in $\mathcal{L}_{\text{PM}}$ largely enhances the rotation accuracy. Nonetheless, the overall performance is slightly inferior to our disentangled formulation $\mathcal{L}_{\text{Pose}}$, which disentangles $\mathbf{t}_{\text{SITE}}$ rather than the 3D translation $\mathbf{t}$. It is worth noting that $\mathcal{L}_{\mathbf{R},\text{sym}}$ has a rather insignificant contribution compared with $\mathcal{L}_{\mathbf{R}}$. This can be accounted to the lack of severe symmetries in LM and to our proposed surface region attention M_{SRA}.

Effectiveness of Geometry-Guided Direct Regression. Furthermore, we train GDR-Net leveraging only our pose loss $\mathcal{L}_{\text{Pose}}$ by discarding the geometric supervision $\mathcal{L}_{\text{Geom}}$. Surprisingly, even the simple version outperforms CDPN [28] w.r.t. ADD(-S) 0.1d, when employing $\mathbf{R}_{\text{6d}}$ for rotation (Tab. 1b G0 *vs*. A0). Yet, we clearly outperform our baseline using GDR-Net with explicit geometric guidance. If we predict the rotation as allocentric quaternions, the accuracy decreases (G2 *vs*. G0), which can partially account for the weak performance of previous direct methods [60, 11]. Moreover, when we remove the guidance of M_{2D}, the accuracy drops significantly (G0 *vs*. G1). Based on these results, we can see that an appropriate geometric guidance is essential for direct 6D pose regression.

Direct pose regression also enhances the learning of geometric features as the error signal from pose can be back-propagated. Tab. 1b (B1, B2) shows that when evaluating GDR-Net with PnP/RANSAC from the predicted $M_{\text{2D-3D}}$, the overall performance exceeds CDPN [28]. Similar to CDPN, we run tests using PnP/RANSAC for $\mathbf{R}$ and Patch-PnP for $\mathbf{t}$, which achieves the overall best accuracy (B2).

Method	w/o Refinement					w/ Refinement	
	PoseCNN [60]	PVNet [40]	Single-Stage [19]	HybridPose [47]	GDR-Net (**Ours**)	DPOD [62]	DeepIM [27]
Ape	9.6	15.8	19.2	20.9	46.8	-	59.2
Can	45.2	63.3	65.1	75.3	90.8	-	63.5
Cat	0.9	16.7	18.9	24.9	40.5	-	26.2
Driller	41.4	65.7	69.0	70.2	82.6	-	55.6
Duck	19.6	25.2	25.3	27.9	46.9	-	52.4
Eggbox*	22.0	50.2	52.0	52.4	54.2	-	63.0
Glue*	38.5	49.6	51.4	53.8	75.8	-	71.7
Holepuncher	22.1	36.1	45.6	54.2	60.1	-	52.5
MEAN	24.9	40.8	43.3	47.5	**62.2**	47.3	55.5

Table 2: **Comparison with State of the Art on LM-O.** We report the Average Recall (%) of ADD(-S). (*) denotes symmetric objects.

Method	Ref.	ADD(-S)	AUC of ADD-S	AUC of ADD(-S)
PoseCNN [60]		21.3	75.9	61.3
SegDriven [20]		39.0	-	-
PVNet [40]		-	-	73.4
Single-Stage [19]		53.9	-	-
GDR-Net (**Ours**)		**60.1**	**91.6**	**84.4**
DeepIM [27]	✓	-	88.1	81.9
CosyPose [25]	✓	-	89.8	**84.5**

Table 3: **Comparison with State of the Art on YCB-V.** We report the results evaluated w.r.t. ADD(-S), and AUC of ADD-S and ADD(-S). As in [60], ADD-S uses the symmetric metric for all objects, while ADD(-S) only uses the symmetric metric for symmetric objects. Ref. stands for Refinement and "-" denotes unavailable results.

This demonstrates that our unified GDR-Net can leverage the best of both worlds, namely, geometry-based indirect methods and direct methods.

Effectiveness of Detection and Pose Decoupling. Similar to CDPN [28], we decouple the detector and GDR-Net by means of Dynamic Zoom-In (DZI). When evaluating GDR-Net with the Yolov3 detections from [28], the overall accuracy only drops slightly while the accuracy for ADD(-S) 0.1d almost remains unchanged (Tab. 1b H0).

4.4. Comparison with State of the Art

We compare our approach with state-of-the-art methods on the LM-O and YCB-V datasets. During training, we apply similar color augmentation as in [49] to prevent overfitting. For the sake of fairness, we train one GDR-Net per object on these two datasets following [40, 20, 19, 47]. For YCB-V, due to the large number of symmetric objects, the symmetric variant for the pose loss $\mathcal{L}_{\mathbf{R},\text{sym}}$ is employed. During testing, for LM-O, we employ Faster-RCNN [44] to obtain 2D detections from the RGB images; for YCB-V, we utilize the publicly available detections from FCOS [52] [2].

Results on LM-O. Tab. 2 presents the results of GDR-Net compared with state-of-the-art methods on LM-O. Our GDR-Net significantly surpasses state-of-the-art methods without refinement such as HybridPose [47] and even outperforms the refinement-based method DeepIM [27].

Results on YCB-V. We compare GDR-Net to state-of-the-art approaches on YCB-V in Tab. 3 (Please refer to the supplement for more detailed results). Our GDR-Net exceeds again state of the art, even without leveraging any refinement. Moreover, our GDR-Net is comparable to the state-of-the-art refinement-based method CosyPose [25] w.r.t. AUC of ADD(-S) metric. Note that our approach runs much faster than the methods requiring refinement.

4.5. Runtime Analysis

On a desktop with an Intel 3.40GHz CPU and an NVIDIA 2080Ti GPU, given a 640×480 image, using the Yolov3 [43] detector, our approach takes $\approx$ 22ms for a single object and $\approx$ 35ms for 8 objects, including 15ms for detection.

5. Conclusion

In this work, we revisited the ingredients of direct 6D pose regression and proposed a novel GDR-Net to unify direct and geometry-based indirect methods. The key idea is to exploit the intermediate geometric features regarding 2D-3D correspondences organized regularly as image-like 2D patches, which facilitates us to utilize a simple yet effective 2D convolutional Patch-PnP to directly regress 6D pose from geometric guidance. Our approach achieves real-time, accurate and robust monocular 6D object pose estimation. In the future, we want to extend our work to more challenging scenarios, such as the lack of annotated real data [55, 34] and unseen object categories or instances [56, 37].

[2] https://github.com/LZGMatrix/BOP19_CDPN_2019ICCV

References

[1] Deniz Beker, Hiroharu Kato, Mihai Adrian Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Monocular Differentiable Rendering for Self-Supervised 3D Object Detection. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*, 2020. 5

[3] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D Object Pose Estimation Using 3D Object Coordinates. In *European Conference on Computer Vision (ECCV)*, pages 536–551, 2014. 2, 4, 5, 12, 15

[4] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC-Differentiable RANSAC for Camera Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6684–6692, 2017. 2

[5] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3364–3372, 2016. 5

[6] Eric Brachmann and Carsten Rother. Learning Less is More-6D Camera Localization via 3D Surface Regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4654–4662, 2018. 2

[7] Eric Brachmann and Carsten Rother. Neural-Guided RANSAC: Learning Where to Sample Model Hypotheses. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4322–4331, 2019. 2

[8] 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 (CVPR)*, pages 8100–8109, 2020. 2

[9] Dengsheng Chen, Jun Li, Zheng Wang, and Kai Xu. Learning Canonical Shape Space for Category-level 6D Object Pose and Size Estimation. In *CVPR*, pages 11973–11982, 2020. 3

[10] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The MOPED Framework: Object Recognition and Pose Estimation for Manipulation. *The International Journal of Robotics Research*, 30(10):1284–1306, 2011. 1

[11] Thanh-Toan Do, Trung Pham, Ming Cai, and Ian Reid. LieNet: Real-time Monocular Object Instance 6D Pose Estimation. In *British Machine Vision Conference (BMVC)*, 2018. 2, 3, 7

[12] Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. Gradient Response Maps for Real-time Detection of Textureless Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(5):876–888, 2012. 1

[13] 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 (ACCV)*, pages 548–562, 2012. 1, 2, 4, 5, 6, 12, 14

[14] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. Going Further with Point Pair Features. In *European Conference on Computer Vision (ECCV)*, pages 834–848, 2016. 1, 2

[15] Tomas Hodan, Daniel Barath, and Jiri Matas. EPOS: Estimating 6D Pose of Objects with Symmetries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11703–11712, 2020. 1, 2, 5

[16] Tomáš Hodaň, Jiří Matas, and Štěpán Obdržálek. On Evaluation of 6D Object Pose Estimation. *European Conference on Computer Vision Workshops (ECCVW)*, pages 606–619, 2016. 6

[17] 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 *European Conference on Computer Vision (ECCV)*, pages 19–34, 2018. 2

[18] Tomas Hodan, Martin Sundermeyer, Bertram Drost, Yann Labbe, Eric Brachmann, Frank Michel, Carsten Rother, and Jiri Matas. BOP Challenge 2020 on 6D Object Localization. *European Conference on Computer Vision Workshops (ECCVW)*, 2020. 1, 2, 6

[19] 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. 2, 5, 6, 8, 12

[20] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6D Object Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3385–3394, 2019. 2, 8, 12

[21] Frank Hutter Ilya Loshchilov. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations (ICLR)*, 2017. 5

[22] 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 *IEEE International Conference on Computer Vision (ICCV)*, pages 1521–1529, 2017. 1, 2, 4

[23] Steven G Krantz and Harold R Parks. *The Implicit Function Theorem: History, Theory, and Applications*. Springer Science & Business Media, 2012. 2

[24] Abhijit Kundu, Yin Li, and James M Rehg. 3D-RCNN: Instance-level 3D Object Reconstruction via Render-and-Compare. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3559–3568, 2018. 4

[25] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent Multi-view Multi-object 6D Pose Estimation. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 3, 4, 5, 8, 13

[26] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision (IJCV)*, 81(2):155, 2009. 5, 6, 12

[27] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep Iterative Matching for 6D Pose Estimation. *International Journal of Computer Vision (IJCV)*, pages 1–22, 2019. 2, 3, 4, 6, 7, 8, 12, 13

[28] Zhigang Li, Gu Wang, and Xiangyang Ji. CDPN: Coordinates-Based Disentangled Pose Network for Real-Time RGB-Based 6-DoF Object Pose Estimation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 7678–7687, 2019. 1, 2, 4, 5, 6, 7, 8, 12

[29] 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 (ICLR)*, April 2020. 5

[30] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking Pseudo-Lidar Representation. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[31] Fabian Manhardt, Diego 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 *IEEE International Conference on Computer Vision (ICCV)*, pages 6841–6850, 2019. 1, 2

[32] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. ROI-10D: Monocular Lifting of 2D Detection to 6D Pose and Metric Shape. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2069–2078, 2019. 1

[33] Fabian Manhardt, Wadim Kehl, Nassir Navab, and Federico Tombari. Deep Model-based 6D Pose Refinement in RGB. In *European Conference on Computer Vision (ECCV)*, pages 800–815, 2018. 2, 3

[34] Fabian Manhardt, Gu Wang, Benjamin Busam, Manuel Nickel, Sven Meier, Luca Minciullo, Xiangyang Ji, and Nassir Navab. CPS++: Improving Class-level 6D Pose and Shape Estimation From Monocular Images With Self-Supervised Learning. *arXiv preprint arXiv:2003.05848*, 2020. 2, 3, 8

[35] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose Estimation for Augmented Reality: a Hands-on Survey. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 22(12):2633–2651, 2015. 1

[36] Frank Michel, Alexander Kirillov, Erix Brachmann, Alexander Krull, Stefan Gumhold, Bogdan Savchynskyy, and Carsten Rother. Global Hypothesis Generation for 6D Object Pose Estimation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[37] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. LatentFusion: End-to-End Differentiable Reconstruction and Rendering for Unseen Object Pose Estimation. In *CVPR*, pages 10710–10719, 2020. 3, 7, 8

[38] Kiru Park, Timothy Patten, and Markus Vincze. Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 7668–7677, 2019. 2

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An Imperative Style, High-performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8026–8037, 2019. 5

[40] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-wise Voting Network for 6DoF Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4561–4570, 2019. 2, 8, 13

[41] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017. 2, 6

[42] 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 *IEEE International Conference on Computer Vision (ICCV)*, pages 3828–3836, 2017. 1, 2

[43] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*, 2018. 5, 7, 8

[44] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 3, 5, 7, 8

[45] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. 6

[46] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel Lopez-Antequera, and Peter Kontschieder. Disentangling Monocular 3D Object Detection. In *ICCV*, 2019. 4, 7

[47] 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 (CVPR)*, pages 431–440, 2020. 8

[48] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[49] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D Orientation Learning for 6D Object Detection from RGB Images. In *European Conference on Computer Vision (ECCV)*, pages 699–715, 2018. 1, 2, 4, 8

[50] Fulin Tang, Yihong Wu, Xiaohui Hou, and Haibin Ling. 3D Mapping and 6D Pose Computation for Real Time Augmented Reality on Cylindrical Objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. 1

[51] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 292–301, 2018. 2

[52] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully Convolutional One-Stage Object Detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 9627–9636, 2019. 3, 5, 8

[53] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects. In *Conference on Robot Learning (CoRL)*, pages 306–316, 2018. 1

[54] Joel Vidal, Chyi-Yeu Lin, Xavier Lladó, and Robert Martí. A Method for 6D Pose Estimation of Free-form Rigid Objects Using Point Pair Features on Range Data. *Sensors*, 18(8):2678, 2018. 1, 2

[55] Gu Wang, Fabian Manhardt, Jianzhun Shao, Xiangyang Ji, Nassir Navab, and Federico Tombari. Self6D: Self-Supervised Monocular 6D Object Pose Estimation. In *European Conference on Computer Vision (ECCV)*, August 2020. 2, 8

[56] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation. In *CVPR*, 2019. 3, 5, 8

[57] Paul Wohlhart and Vincent Lepetit. Learning Descriptors for Object Recognition and 3D Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3109–3118, 2015. 2

[58] Di Wu, Zhaoyong Zhuang, Canqun Xiang, Wenbin Zou, and Xia Li. 6D-VNet: End-To-End 6-DoF Vehicle Pose Estimation From Monocular RGB Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2019. 1

[59] Yuxin Wu and Kaiming He. Group Normalization. In *European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 4

[60] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. *Robotics: Science and Systems (RSS)*, 2018. 2, 3, 4, 5, 6, 7, 8, 12, 13, 16

[61] 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 (ECCV)*, 2020. 5

[62] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. DPOD: 6D Pose Object Detector and Refiner. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1941–1950, 2019. 2, 8

[63] Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker, Alberto Rodriguez, and Jianxiong Xiao. Multi-view Self-supervised Deep Learning for 6D Pose Estimation in the Amazon Picking Challenge. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1386–1383, 2017. 1, 2

[64] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead Optimizer: k Steps Forward, 1 Step Back. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9593–9604, 2019. 5

[65] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the Continuity of Rotation Representations in Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5745–5753, 2019. 3

[66] Menglong Zhu, Konstantinos G Derpanis, Yinfei Yang, Samarth Brahmbhatt, Mabel Zhang, Cody Phillips, Matthieu Lecce, and Kostas Daniilidis. Single Image 3D Object Detection and Pose Estimation for Grasping. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3936–3943. IEEE, 2014. 1

Method	PoseCNN [60]	SegDriven [20]	Single-Stage [19]	GDR-Net (**Ours**)
002_master_chef_can	3.6	33.0	-	41.5
003_cracker_box	25.1	44.6	-	83.2
004_sugar_box	40.3	75.6	-	91.5
005_tomato_soup_can	25.5	40.8	-	65.9
006_mustard_bottle	61.9	70.6	-	90.2
007_tuna_fish_can	11.4	18.1	-	44.2
008_pudding_box	14.5	12.2	-	2.8
009_gelatin_box	12.1	59.4	-	61.7
010_potted_meat_can	18.9	33.3	-	64.9
011_banana	30.3	16.6	-	64.1
019_pitcher_base	15.6	90.0	-	99.0
021_bleach_cleanser	21.2	70.9	-	73.8
024_bowl*	12.1	30.5	-	37.7
025_mug	5.2	40.7	-	61.5
035_power_drill	29.9	63.5	-	78.5
036_wood_block*	10.7	27.7	-	59.5
037_scissors	2.2	17.1	-	3.9
040_large_marker	3.4	4.8	-	7.4
051_large_clamp*	28.5	25.6	-	69.8
052_extra_large_clamp*	19.6	8.8	-	90.0
061_foam_brick*	54.5	34.7	-	71.9
MEAN	21.3	39.0	53.9	**60.1**

Table B.1: **Detailed results on YCB-V [60] w.r.t. ADD(-S).** Baseline approaches are PoseCNN [60], SegDriven [20], and Single-Stage [19]. (*) denotes symmetric objects.

A. Details of PnP/RANSAC

The implementation and hyper-parameters of PnP/RANSAC follow the state-of-the-art method CDPN [28] for all our experiments. Specifically, we leverage EPnP [26] together with 100 RANSAC iterations using a reprojection error threshold of 3 and confidence threshold of 0.99.

B. Detailed Results of YCB-V

We present detailed evaluation results on YCB-V [60] for our GDR-Net in Tab. B.1 and Tab. C.2 and compare them to state-of-the-art approaches w.r.t. ADD(-S) and AUC of ADD-S/ADD(-S), respectively. As for methods trained simultaneously for all objects, our GDR-Net clearly outperforms all other state-of-the-art methods. Furthermore, when GDR-Net is trained separately for each individual object, we can even surpass refinement-based methods such as DeepIM [27] w.r.t. AUC of ADD-S/ADD(-S) metric.

C. Qualitative Results

We demonstrated additional qualitative results for LM [13], LM-O [3], and YCB-V [60] in Fig. C.1, Fig. C.2 and Fig. C.3, respectively. Thereby, in Fig. C.1, we visualize the 6D pose by overlaying the image with the corresponding transformed 3D bounding box. In Fig. C.2 and Fig. C.3, we illustrate the estimated 6D poses by rendering the 3D models on top of the input image and highlighting the respective contours. Note that while *Blue* constitutes the ground-truth poses, we demonstrate in *Green* the predicted poses from GDR-Net. For better visualization we cropped the images and zoomed into the area of interest.

Method	w/o Refinement					w/ Refinement			
	PoseCNN [60]		PVNet [40]	GDR-Net (**Ours**)		DeepIM [27]		CosyPose [25]	
Metric	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)
002_master_chef_can	84.0	50.9	81.6	96.3	65.2	93.1	71.2	-	-
003_cracker_box	76.9	51.7	80.5	97.0	88.8	91.0	83.6	-	-
004_sugar_box	84.3	68.6	84.9	98.9	95.0	96.2	94.1	-	-
005_tomato_soup_can	80.9	66.0	78.2	96.5	91.9	92.4	86.1	-	-
006_mustard_bottle	90.2	79.9	88.3	100.0	92.8	95.1	91.5	-	-
007_tuna_fish_can	87.9	70.4	62.2	99.4	94.2	96.1	87.7	-	-
008_pudding_box	79.0	62.9	85.2	64.6	44.7	90.7	82.7	-	-
009_gelatin_box	87.1	75.2	88.7	97.1	92.5	94.3	91.9	-	-
010_potted_meat_can	78.5	59.6	65.1	86.0	80.2	86.4	76.2	-	-
011_banana	85.9	72.3	51.8	96.3	85.8	91.3	81.2	-	-
019_pitcher_base	76.8	52.5	91.2	99.9	98.5	94.6	90.1	-	-
021_bleach_cleanser	71.9	50.5	74.8	94.2	84.3	90.3	81.2	-	-
024_bowl*	69.7	69.7	89.0	85.7	85.7	81.4	81.4	-	-
025_mug	78.0	57.7	81.5	99.6	94.0	91.3	81.4	-	-
035_power_drill	72.8	55.1	83.4	97.5	90.1	92.3	85.5	-	-
036_wood_block*	65.8	65.8	71.5	82.5	82.5	81.9	81.9	-	-
037_scissors	56.2	35.8	54.8	63.8	49.5	75.4	60.9	-	-
040_large_marker	71.4	58.0	35.8	88.0	76.1	86.2	75.6	-	-
051_large_clamp*	49.9	49.9	66.3	89.3	89.3	74.3	74.3	-	-
052_extra_large_clamp*	47.0	47.0	53.9	93.5	93.5	73.3	73.3	-	-
061_foam_brick*	87.8	87.8	80.6	96.9	96.9	81.9	81.9	-	-
MEAN	75.9	61.3	73.4	**91.6**	**84.3**	88.1	81.9	89.8	**84.5**

Table C.2: **Detailed results on YCB-V [60] w.r.t. AUC of ADD-S and ADD(-S).** (*) denotes symmetric objects.

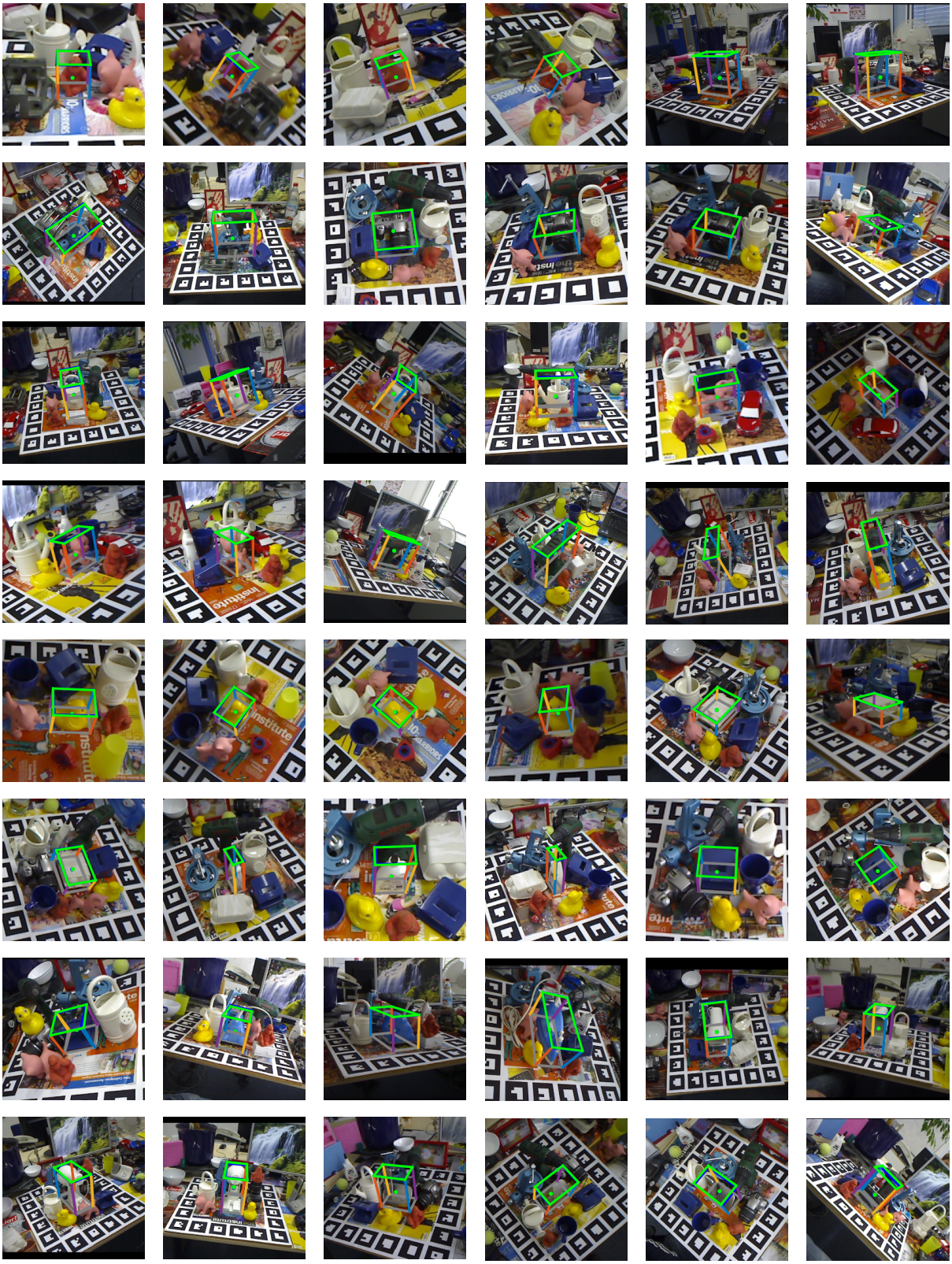


Figure C.1: **Qualitative Results on LM [13].** We visualize the 6D pose by overlaying the image with the corresponding transformed 3D bounding box. We demonstrate in *Blue* and *Green* the ground-truth pose and the predicted pose, respectively.

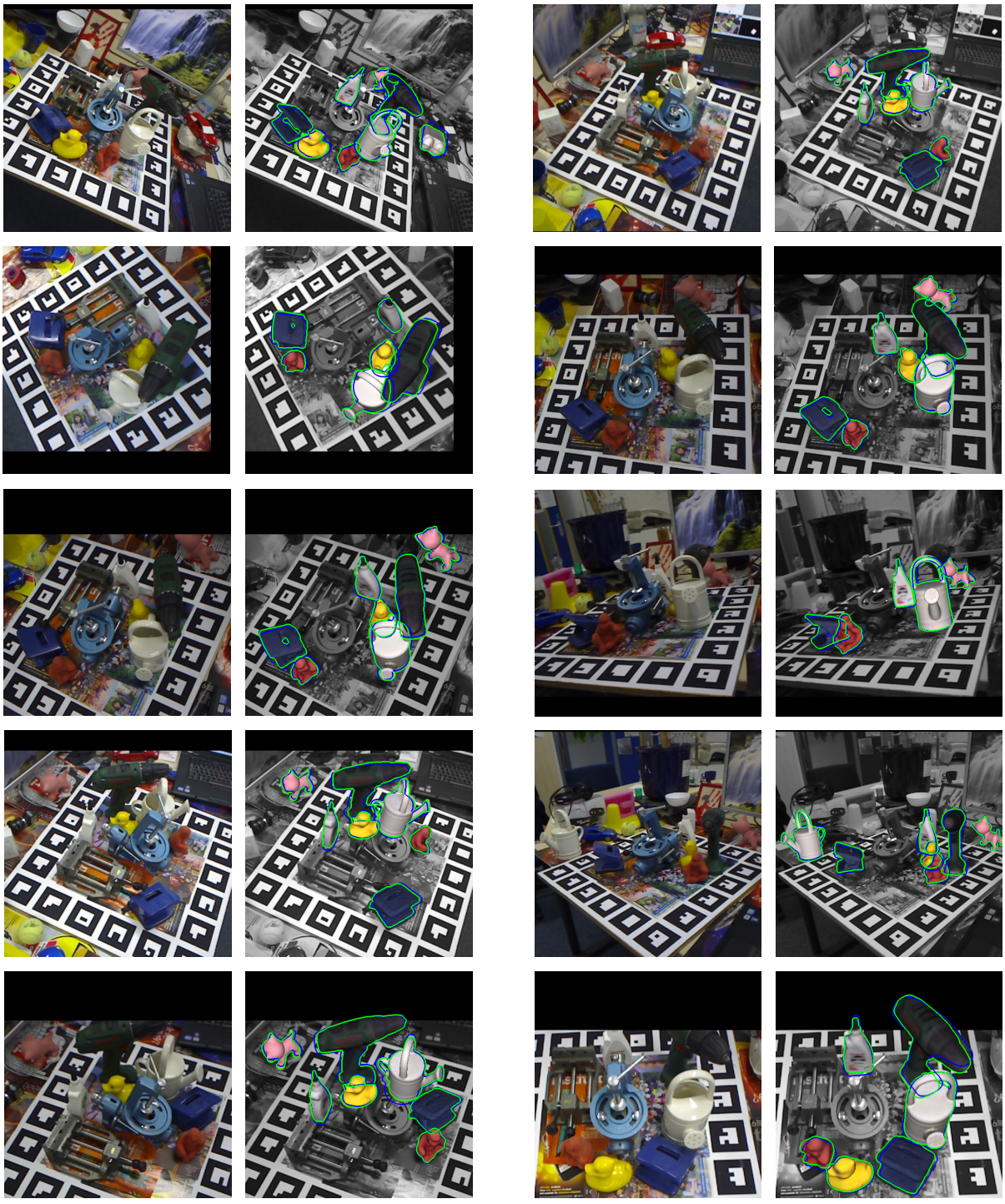


Figure C.2: **Qualitative Results on LM-O [3].** For each image, we visualize the 6D poses by rendering the 3D models and overlaying the contours on the right. We demonstrate in *Blue* and *Green* the ground-truth pose and the predicted pose, respectively.

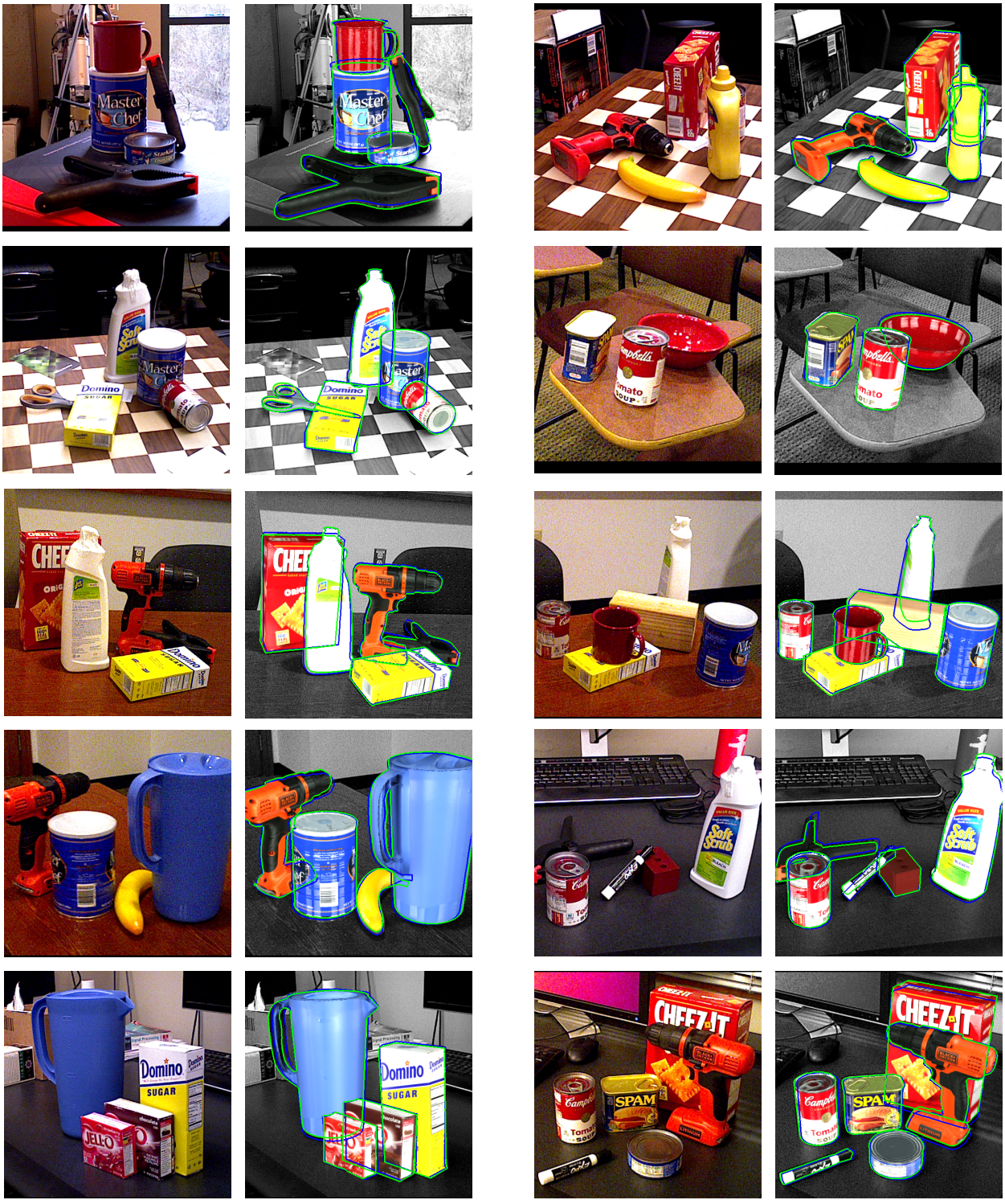


Figure C.3: **Qualitative Results on YCB-V [60].** For each image, we visualize the 6D poses by rendering the 3D models and overlaying the contours on the right. We demonstrate in *Blue* and *Green* the ground-truth pose and the predicted pose, respectively.