BOP Challenge 2023 on Detection, Segmentation and Pose Estimation of Seen and Unseen Rigid Objects

Tomas Hodan[1] Martin Sundermeyer[2] Yann Labbé[1] Van Nguyen Nguyen[3] Gu Wang[4]
Eric Brachmann[5] Bertram Drost[6] Vincent Lepetit[3] Carsten Rother[7] Jiri Matas[8]

[1]Meta [2]Google [3]ENPC [4]Tsinghua University [5]Niantic [6]MVTec [7]Heidelberg University [8]CTU in Prague

Abstract

We present the evaluation methodology, datasets and results of the BOP Challenge 2023, the fifth in a series of public competitions organized to capture the state of the art in model-based 6D object pose estimation from an RGB/RGB-D image and related tasks. Besides the three tasks from 2022 (2D detection, 2D segmentation, and 6D localization of objects seen during training), the 2023 challenge introduced new variants of these tasks focused on objects unseen during training. In the new tasks, methods were required to learn new objects during a short onboarding stage (max 5 minutes, 1 GPU) from provided 3D object models. The best 2023 method for 6D localization of unseen objects (GenFlow) notably reached the accuracy of the best 2020 method for seen objects (CosyPose), although being noticeably slower. The best 2023 method for seen objects (GPose) achieved a moderate accuracy improvement but a significant 43% run time improvement compared to the best 2022 counterpart (GDRNPP). Since 2017, the accuracy of 6D localization of seen objects has improved by more than 50% (from 56.9 to 85.6 AR_C). The online evaluation system stays open and is available at: `bop.felk.cvut.cz`.

1. Introduction

The BOP Challenge 2023 was the fifth in a series of public challenges that are part of the BOP[1] project, which aims to continuously record and report the state of the art in estimating the 6D object pose (3D translation and 3D rotation) and related tasks such as 2D object detection and segmentation. Results of the previous editions of the challenge from 2017, 2019, 2020, and 2022 were published in [21, 24, 25, 55].

Participants of the 2023 challenge were competing on six tasks. Besides the three tasks from 2022 (model-based 2D object detection, 2D object segmentation and 6D object localization of objects seen during training), the 2023 challenge introduced new variants of these tasks focused on *objects unseen during training*. In the new tasks, methods were required to adapt to novel 3D object models during a short object onboarding stage (max 5 min per object, 1 GPU), and then recognize the objects in images from diverse environments. Such methods are of high practical relevance as they do not require expensive data generation and training for every new object, which is typically required by most existing methods for seen objects and severely limits their scalability. The introduction of the new tasks was encouraged by the recent breakthroughs in foundation models and their impressive few-shot learning capabilities.

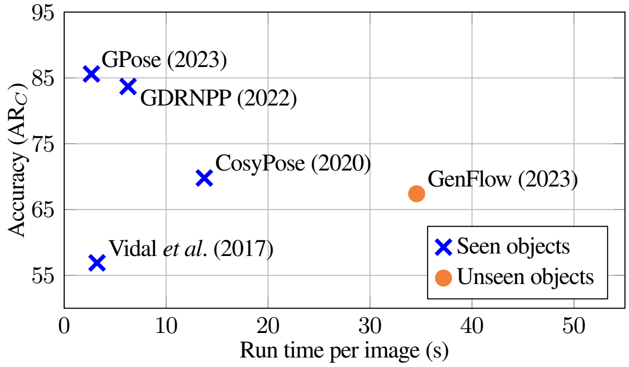

Figure 1. **Progress in model-based 6D object localization (2017–2023).** Shown is the accuracy and run time of the top performing RGB-D methods on the seven core BOP datasets. The dominance of methods based on point-pair features [10], represented by Vidal *et al.* [60] in 2017, was ended by the learning-based CosyPose [32] in 2020 for the price of a significantly higher run time. In 2022, GDRNPP [39, 61] dramatically improved both accuracy and run time. Finally, in 2023, GPose [67] brought the run time back to the 2017 level while further improving the accuracy. The field has come a long way since 2017 – the accuracy has improved by more than 50% (from 56.9 to 85.6 AR_C). GenFlow [40], the best method for the newly introduced task of 6D localization of *unseen objects* (objects not seen during training), reaches the accuracy of CosyPose, the best 2020 method for *seen objects*, while its run time awaits improvements.

The challenge primarily focuses on the practical scenario where no real images are available at training/onboarding time, only the 3D object models and images synthesized using the models. While capturing real images of objects under various conditions and annotating the images with 6D object poses requires a significant human effort [22], the 3D models are either available before the physical objects, which is often the case for manufactured objects, or can be reconstructed at an admissible cost. Approaches for reconstructing 3D models of opaque, matte and moderately specular objects are established [42, 49] and promising approaches for transparent and highly specular objects are emerging [14, 41, 59, 62].

[1] BOP stands for Benchmark for 6D Object Pose Estimation [24].

In the 2019 challenge, methods using the depth image channel were mostly based on point pair features (PPF's) [10] and clearly outperformed methods relying only on the RGB channels, all of which were based on deep neural networks (DNN's). DNN-based methods need large amounts of annotated training images, which had been typically obtained by OpenGL rendering of the 3D object models on random backgrounds [18, 30]. However, as suggested in [26], the evident domain gap between these "render & paste" training images and real test images limits the potential of the DNN-based methods. To reduce the gap between the synthetic and real domains and thus to bring fresh air to the DNN world, we joined the development of BlenderProc[2] [4, 5], an open-source, physically-based renderer (PBR). For the 2020 challenge, we then provided participants with 350K PBR training images (see [25] for examples), which helped the DNN-based methods to achieve noticeably higher accuracy and to finally catch up with the PPF-based methods. In the 2022 challenge, DNN-based methods for 6D object localization already clearly outperformed PPF-based methods in both accuracy and speed, with the performance gains coming mostly from advances in network architectures and training schemes.

Remarkably, RGB methods from 2022 surpassed RGB-D methods from 2020, the performance gap between methods trained only on PBR images and methods trained also on real images noticeably shrank, and some methods started training on the depth image channel in addition to the RGB channels. In 2022, we started evaluating also the tasks of 2D object detection and 2D object segmentation, to address the design of the majority of recent object pose estimation methods, which start by detecting/segmenting objects and then estimate their poses from the predicted image regions. Evaluating the detection/segmentation and pose estimation stages separately enabled a better understanding of the progress in object pose estimation.

In 2023, we introduced three more practical tasks focused on unseen objects, *i.e*. the target objects are not seen during training and need to be onboarded with limited resources (max 5 minutes on 1 GPU). While similar tasks have been considered in the literature [33, 44, 52], direct comparison of methods has been difficult due to variations in the detection stage and the used training data. To address this situation, we proposed a unified evaluation framework utilizing an open-source detection method and a large-scale training dataset. Specifically, CNOS [43], a model-based method for detecting/segmenting unseen objects that outperforms Mask-RCNN [16], was employed as the default method for 2D detection and segmentation. As the training dataset, we used synthetic training data from MegaPose [33]. Methods were not required but encouraged (via dedicated awards) to use these unified solutions.

The best 2023 method for 6D localization of unseen objects (GenFlow [40]) reached the accuracy of the best 2020 method for seen objects (CosyPose [32]). Despite being noticeably slower, this is an impressive result considering that the target objects are onboarded in a short time, which is several orders of magnitude shorter than a typical training process of methods trained for specific objects. The best 2023 method for seen objects (GPose [67]) achieves a moderate accuracy improvement and a significant 42.6% run time improvement compared to the best 2022 counterpart (GDRNPP [39, 61]).

Sec. 2 of this report defines the evaluation methodology, Sec. 3 introduces datasets, Sec. 4 describes the experimental setup and analyzes the results, Sec. 5 presents the awards of the BOP Challenge 2023, and Sec. 6 concludes the report.

2. Challenge tasks

Methods are evaluated on the task of model-based 6D localization on seen objects (as in 2019, 2020 and 2022 [55]), on the tasks of model-based 2D detection and 2D segmentation of seen objects (as in 2022 [55]), and on variants of these tasks focused on objects unseen during training, which were introduced in 2023. All six tasks are defined below, together with accuracy scores that are used to compare methods. Participants could submit their results to any of the six tasks. Note that although all BOP datasets currently include RGB-D images (Sec. 3), a method may have used any of the image channels.

2.1. Task 1: 6D localization of seen objects

The definition of this task is the same since 2019, which enables direct comparison across the years[3].

Training input: At training time, a method is provided a set of RGB-D training images showing objects annotated with ground-truth 6D poses, and 3D mesh models of the objects (typically with a color texture). A 6D pose is defined by a matrix $\mathbf{P} = [\mathbf{R}|\mathbf{t}]$, where $\mathbf{R}$ is a 3D rotation matrix, and $\mathbf{t}$ is a 3D translation vector. The matrix $\mathbf{P}$ defines a rigid transformation from the 3D space of the object model to the 3D space of the camera.

Test input: At test time, the method is given an RGB-D image unseen during training and a list $L = [(o_1, n_1), \ldots, (o_m, n_m)]$, where n_i is the number of instances of object o_i visible in the image. In 2023, methods could use provided default detections (results of GDRNPPDet_PBRReal, the best 2D detection method from 2022 for Task 2).

Test output: The method produces a list $E = [E_1, \ldots, E_m]$, where E_i is a list of n_i pose estimates with confidences for instances of object o_i.

Evaluation methodology: The error of an estimated pose w.r.t. the ground-truth pose is calculated by three pose-error functions (see Sec. 2.2 of [25] for details): (1) VSD (Visible Surface Discrepancy) which treats indistinguishable poses as equivalent by considering only the visible object part, (2) MSSD (Maximum Symmetry-Aware Surface Distance) which considers a set of pre-identified global object symmetries and measures the surface deviation in 3D, (3) MSPD (Maximum Symmetry-Aware Projection Distance) which considers the object symmetries and measures the perceivable deviation.

[2] github.com/DLR-RM/BlenderProc

[3] See Sec. A.1 in [25] for a discussion on why the methods are evaluated on 6D object localization instead of 6D object detection, where no prior information about the visible object instances is provided [23].

An estimated pose is considered correct w.r.t. a pose-error function e, if $e < \theta_e$, where $e \in \{\text{VSD}, \text{MSSD}, \text{MSPD}\}$ and θ_e is the threshold of correctness. The fraction of annotated object instances for which a correct pose is estimated is referred to as Recall. The Average Recall w.r.t. a function e, denoted as AR_e, is defined as the average of the Recall rates calculated for multiple settings of the threshold θ_e and also for multiple settings of a misalignment tolerance τ in the case of VSD. The accuracy of a method on a dataset D is measured by: $\text{AR}_D = (\text{AR}_{\text{VSD}} + \text{AR}_{\text{MSSD}} + \text{AR}_{\text{MSPD}})/3$, which is calculated over estimated poses of all objects from D. The overall accuracy on the core datasets is measured by AR_C defined as the average of the per-dataset AR_D scores (see Sec. 2.4 of [25] for details)[4].

2.2. Task 2: 2D detection of seen objects

Training input: At training time, a method is provided a set of RGB-D training images showing objects annotated with ground-truth 2D bounding boxes. The boxes are *amodal*, *i.e.*, covering the whole object silhouette, including the occluded parts. The method can use the 3D mesh models that are available for the objects (*e.g.*, to synthesize extra training images).

Test input: At test time, the method is given an RGB-D image unseen during training that shows an arbitrary number of instances of an arbitrary number of objects, with all objects being from one specified dataset (*e.g.* YCB-V [64]). No prior information about the visible object instances is provided.

Test output: The method produces a list of object detections with confidences, with each detection defined by an *amodal* 2D bounding box.

Evaluation methodology: Following the evaluation methodology from the COCO 2020 Object Detection Challenge [36], the detection accuracy is measured by the Average Precision (AP). Specifically, a per-object AP_O score is calculated by averaging the precision at multiple Intersection over Union (IoU) thresholds: $[0.5, 0.55, ..., 0.95]$. The accuracy of a method on a dataset D is measured by AP_D calculated by averaging per-object AP_O scores, and the overall accuracy on the core datasets (Sec. 3) is measured by AP_C defined as the average of the per-dataset AP_D scores. Analogous to the 6D localization task, only annotated object instances for which at least 10% of the projected surface area is visible need to be detected. Correct predictions for instances that are visible from less than 10% are filtered out and not counted as false positives. Up to 100 predictions per image with the highest confidences are considered.

2.3. Task 3: 2D segmentation of seen objects

Training input: At training time, a method is provided a set of RGB-D training images showing objects that are annotated with ground-truth 2D binary masks. The masks are *modal*, *i.e.*, covering only the visible object parts. The method can also use 3D mesh models that are available for the objects.

Test input: At test time, the method is given an RGB-D image unseen during training that shows an arbitrary number of instances of an arbitrary number of objects, with all objects being from one specified dataset (*e.g.* YCB-V). No prior information about the visible object instances is provided.

Test output: The method produces a list of object segmentations with confidences, with each segmentation defined by a *modal* 2D binary mask.

Evaluation methodology: As in Task 2, with the only difference being that IoU is calculated on masks instead of bounding boxes.

2.4. Task 4: 6D localization of unseen objects

Training input: At training time, a method is provided a set of RGB-D training images showing training objects annotated with ground-truth 6D poses, and 3D mesh models of the objects (typically with a color texture). The 6D object pose is defined as in Task 1. The method can use 3D mesh models that are available for the training objects.

Object-onboarding input: The method is provided 3D mesh models of test objects that were not seen during training. To onboard each object (*e.g.* to render images/templates or fine-tune a neural network), the method can spend up to 5 minutes of the wall-clock time on a computer with a single GPU. The time is measured from the point right after the raw data (*e.g.* 3D mesh models) is loaded to the point when the object is onboarded. The method can render images of the 3D object models but cannot use any real images of the objects for onboarding. The object representation (which may be given by a set of templates, a machine-learning model, *etc.*) needs to be fixed after onboarding (it cannot be updated on test images).

Test input: At test time, the method is given an RGB-D image unseen during training and a list $L = [(o_1, n_1), ..., (o_m, n_m)]$, where n_i is the number of instances of object o_i visible in the image. In 2023, the method can use provided default detections/segmentations produced by CNOS [43].

Test output: As in Task 1.

Evaluation methodology: As in Task 1.

2.5. Task 5: 2D detection of unseen objects

Training input: At training time, a method is provided a set of RGB-D training images showing training objects that are annotated with ground-truth 2D bounding boxes. The boxes are *amodal*, *i.e.*, covering the whole object silhouette including the occluded parts. The method can also use 3D mesh models that are available for the training objects.

Object-onboarding input: As in Task 4.

Test input: At test time, the method is given an RGB-D image unseen during training that shows an arbitrary number of instances of an arbitrary number of test objects, with all objects being from one specified dataset (*e.g.* YCB-V). No prior information about the visible object instances is provided.

[4] When calculating AR_C, scores are not averaged over objects before averaging over datasets, which is done when calculating AP_C (Sec. 2.2) to comply with the original COCO evaluation methodology [36].

Test output: As in Task 2.

Evaluation methodology: As in Task 2.

2.6. Task 6: 2D segmentation of unseen objects

Training input: At training time, a method is provided a set of RGB-D training images showing training objects that are annotated with ground-truth 2D binary masks. The masks are *modal*, *i.e.*, covering only the visible object parts. The method can also use 3D mesh models that are available for the training objects.

Object-onboarding input: As in Task 4.

Test input: As in Task 5.

Test output: As in Task 3.

Evaluation methodology: As in Task 3.

3. Datasets

3.1. Core datasets

BOP currently includes twelve datasets in a unified format. Sample test images are in Fig. 2 and dataset parameters in Tab. 1. Seven from the twelve were selected as core datasets: LM-O, T-LESS, ITODD, HB, YCB-V, TUD-L, and IC-BIN. Since 2019, methods must be evaluated on all of these core datasets to be considered for the main challenge awards (Sec. 5).

Each dataset includes 3D object models and training and test RGB-D images annotated with ground-truth 6D object poses. The object models are provided in the form of 3D meshes (in most cases with a color texture) which were created manually or using KinectFusion-like systems for 3D reconstruction [42]. While all test images are real, training images may be real and/or synthetic. The seven core datasets include a total of 350K photorealistic PBR (physically-based rendered) training images generated and automatically annotated with BlenderProc [4–6]. Example images, a description of the generation process and an analysis of the importance of PBR training images are in Sec. 3.2 and 4.3 of the 2020 challenge paper [25]. Datasets T-LESS, TUD-L and YCB-V include also real training images, and most datasets additionally include training images obtained by OpenGL rendering of the 3D object models on a black background. Test images were captured in scenes with graded complexity, often with clutter and occlusion. Datasets HB and ITODD include also real validation images – in this case, the ground-truth poses are publicly available only for the validation and not for the test images. The datasets can be downloaded from the BOP website and more details can be found in Chapter 7 of [19].

3.2. Training dataset for tasks on unseen objects

In 2023, as a training dataset for Tasks 4–6 (Sec. 2), we provided over 2M images in the BOP format showing more than 50K diverse objects (Fig. 2). The images were originally synthesized for MegaPose [33] using BlenderProc [4–6]. The objects are from the Google Scanned Objects [8] and ShapeNetCore [2] datasets. Note that symmetry transformations are not available for these objects, but could be identified as described in Sec. 2.3 of [25].

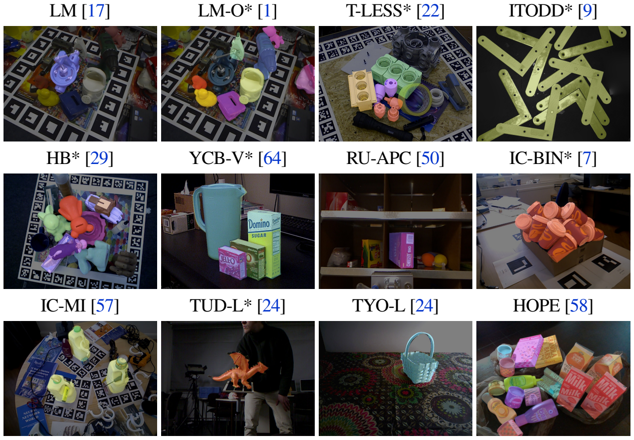

Figure 2. **An overview of the BOP datasets.** The seven core datasets are marked with a star. Shown are RGB channels of sample test images which were darkened and overlaid with colored 3D object models in the ground-truth 6D poses.

		Train. im.		Val im.	Test im.		Test inst.	
Dataset	Obj.	Real	PBR	Real	All	Used	All	Used
LM-O [1]	8	–	50K	–	1214	200	9038	1445
T-LESS [22]	30	37584	50K	–	10080	1000	67308	6423
ITODD [9]	28	–	50K	54	721	721	3041	3041
HB [29]	33	–	50K	4420	13000	300	67542	1630
YCB-V [64]	21	113198	50K	–	20738	900	98547	4123
TUD-L [24]	3	38288	50K	–	23914	600	23914	600
IC-BIN [7]	2	–	50K	–	177	150	2176	1786
LM [17]	15	–	50K	–	18273	3000	18273	3000
RU-APC [50]	14	–	–	–	5964	1380	5964	1380
IC-MI [57]	6	–	–	–	2067	300	5318	800
TYO-L [24]	21	–	–	–	1670	1670	1670	1670
HOPE [58]	28	–	–	50	188	188	3472	2898

Table 1. **Parameters of the BOP datasets.** The core datasets are listed in the upper part. PBR training images rendered by BlenderProc [4, 5] are provided for all core datasets. If a dataset includes both validation and test images, ground-truth annotations are public only for the validation images. All test images are real. Column "Test inst./All" shows the number of annotated object instances for which at least 10% of the projected surface area is visible in the test image. Columns "Used" show the number of used test images and object instances.

Table 2. **Example training images from the MegaPose dataset [33].** This dataset includes 2M images showing annotated instances of more than 50K diverse objects and is meant for training methods for tasks on unseen objects (Tasks 4–6). The objects are not present in any other BOP dataset and their 3D models are available.

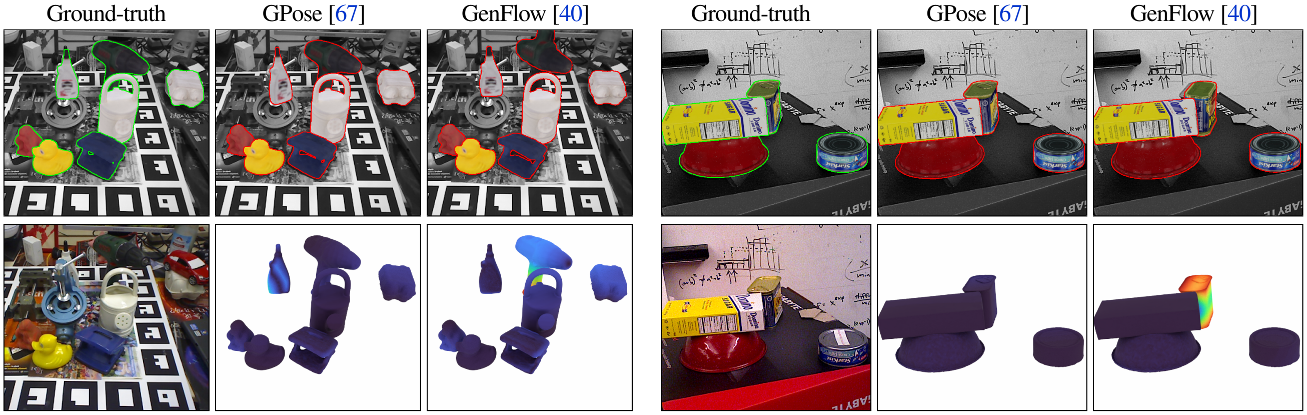

Figure 3. **Qualitative comparison of the state-of-the-art methods for 6D localization of seen (GPose) and unseen objects (GenFlow)** on sample images from LM-O [1] and YCB-V [64]. The bottom row shows the depth error map of each estimated pose w.r.t. the ground-truth pose. The map shows the distance between each 3D point in the ground-truth depth map and its position in the estimated pose (darker red indicates higher error: 0 cm 10 cm). While GenFlow demonstrates strong performance on unseen objects, it tends to fail on challenging cases with heavy object occlusion (*e.g*., the drill in the sample LM-O image or the meat can in the YCB-V image).

4. Results and discussion

This section presents results of the BOP Challenge 2023, compares them with results from earlier challenge editions, and summarizes the main messages for our field. In total, 65 methods were fully evaluated (on all seven core datasets) on Task 1; 9 methods on Task 2; 11 methods on Task 3; 14 methods on Task 4; 3 methods on Task 5 and 4 methods on Task 6. Note that some of the results on Tasks 1–3 are from previous editions of the challenge.

4.1. Experimental setup

Participants of the 2023 challenge were submitting results to the online evaluation system at `bop.felk.cvut.cz` from June 7, 2023 until the deadline on September 28, 2023. The evaluation scripts are publicly available in the BOP toolkit[5].

A method had to use a fixed set of hyper-parameters across all objects and datasets. For the tasks on seen objects (Tasks 1–3), a method could use the provided 3D object models and training images as well as render extra unlimited training images. For the tasks on unseen objects (Tasks 4–6), a method had to onboard new objects from their 3D models in a limited onboarding stage of 5 minutes on a PC with a single GPU. The method could render images of the 3D models or use a subset of the BlenderProc images originally provided for BOP 2020 [25] – the method could use as many images from this set as could be rendered within the limited onboarding time (rendering and any additional processing had to fit within 5 minutes, considering that rendering of one BlenderProc image takes 2 seconds).

Not a single pixel of test images may have been used for training and onboarding, nor the individual ground-truth annotations that are publicly available for test images of some datasets. Ranges of the azimuth and elevation camera angles, and a range of the camera-object distances determined by the ground-truth poses from test images are the only information about the test set that may have been used during training and onboarding. Only subsets of test images were used (see Tab. 1) to remove redundancies and speed up the evaluation, and only object instances for which at least 10% of the projected surface area is visible were considered in the evaluation.

4.2. Results on Task 1

Results on the task of 6D object localization of seen objects and properties of the evaluated methods are in Tab. 3. Among the 16 new entries in 2023, three outperform GDRNPP [39, 61], the best method from the 2022 challenge. The best pose estimation pipeline from 2023, GPose2023 [61, 67], is purely learning-based and achieves 85.6 AR_C, outperforming GDRNPP by 1.9 AR_C (#1−#4 in Tab. 3) with less than half the inference time (2.67 s vs. 6.26 s). GPose2023 deploys the same pose estimation method as GDRNPP but combines it with a more efficient coordinate-guided pose refinement strategy [67] and an improved 2D object detector based on YOLOv8 (see #1−#2 in Tab. 5). Without any pose refinement, the RGB-only variants GPose2023-RGB (#21, 72.9 AR_C) or ZebraPoseSAT-EffnetB4 [53] (#17, 74.9 AR_C) reach an average inference time of ∼0.25 seconds per image which are closer to the demands of mobile vision applications. Gains in accuracy are most notable on the industrial ITODD, T-LESS, and HB datasets, whereas on TUD-L and YCB-V we can observe that metrics start to saturate.

4.3. Results on Tasks 2 and 3

As shown in Tab. 5, GDet2023 [67] based on YOLOv8 [28] achieves 79.8 AP_C, a moderate +2.5 AP_C gain over YOLOX [11], the best detector in 2022. YOLOv8 is even less sensitive to the training image domain than YOLOX, achieving 76.9 AP_C when trained only on synthetic PBR images and neglecting the real training data. In the 2D segmentation of seen objects task (Tab. 6), we see a similar incremental improvement of +3.2 AP_C achieved by ZebraPoseSAT [53], which predicts object masks from the provided default detections of GDRNPP_Det.

[5] `github.com/thodan/bop_toolkit`

#	Method	Year	Type	DNN per	Det./seg.	Refinement	Train im.	...type	Test im.	LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	AR_C	Time
1	GPose2023 [39, 61]	2023	DNN	Object	Custom	∼Coord-guided	RGB-D	PBR+real	RGB-D	79.4	91.4	96.4	73.7	70.4	95.0	92.8	85.6	2.67
2	GPose2023-OfficialDet [39, 61]	2023	DNN	Object	Default GDRNPPDet	∼Coord-guided	RGB-D	PBR+real	RGB-D	80.5	89.5	96.6	73.4	68.7	94.4	92.9	85.1	4.57
3	GPose2023-PBR [39, 61]	2023	DNN	Object	Custom	∼Coord-guided	RGB-D	PBR+real	RGB-D	79.4	89.0	93.1	73.7	70.4	95.0	90.1	84.4	2.86
4	GDRNPP-PBRReal-RGBD-MModel [39, 61]	2022	DNN	Object	YOLOX	∼CIR	RGB-D	PBR+real	RGB-D	77.5	87.4	96.6	72.2	67.9	92.6	92.1	83.7	6.26
5	GDRNPP-PBR-RGBD-MModel [39, 61]	2022	DNN	Object	YOLOX	∼CIR	RGB-D	PBR	RGB-D	77.5	85.2	92.9	72.2	67.9	92.6	90.6	82.7	6.26
6	ZebraPoseSAT-EffnetB4-refined [53]	2023	DNN	Object	Default GDRNPPDet	∼CIR	RGB-D	PBR+real	RGB-D	78.0	86.2	95.6	65.4	61.8	92.1	89.9	81.3	2.57
7	GDRNPP-PBRReal-RGBD-MModel-Fast [39, 61]	2022	DNN	Object	YOLOX	Depth adjust.	RGB	PBR+real	RGB-D	79.2	87.2	93.6	70.2	58.8	90.9	83.4	80.5	0.23
8	OfficialDet-PFA-Mixpbr-RGB-D [27]	2023	DNN	Dataset	Default GDRNPPDet	PFA	RGB	PBR+real	RGB-D	79.2	84.9	96.3	70.6	52.6	86.7	89.9	80.0	1.19
9	GDRNPP-PBRReal-RGBD-MModel-Offi. [39, 61]	2022	DNN	Object	Default (synt+real)	∼CIR	RGB-D	PBR+real	RGB-D	75.8	82.4	96.6	70.8	54.3	89.0	89.6	79.8	6.41
10	GDRNPPDet-PBRReal+GenFlow-MultiHypo [39]	2023	DNN	Dataset	Default (synt+real)	Recurrent Flow	RGB-D	PBR+real	RGB-D	74.4	78.0	92.4	65.1	64.7	91.6	88.4	79.2	36.01
11	Extended_FCOS+PFA-MixPBR-RGBD [27]	2022	DNN	Dataset	Extended FCOS	PFA	RGB	PBR+real	RGB-D	79.7	85.0	96.0	67.6	46.9	86.9	88.8	78.7	2.32
12	Extended_FCOS+PFA-MixPBR-RGBD-Fast [27]	2022	DNN	Dataset	Extended FCOS	PFA	RGB	PBR+real	RGB-D	79.2	77.9	95.8	67.1	46.0	86.0	88.0	77.1	0.64
13	RCVPose3D-SingleModel-VIVO-PBR [63]	2022	DNN	Dataset	RCVPose3D	ICP	RGB-D	PBR+real	RGB-D	72.9	70.8	96.6	73.3	53.6	86.3	84.3	76.8	1.34
14	ZebraPoseSAT-EffnetB4+ICP(DefaultDet) [53]	2022	DNN	Object	Default (synt+real)	ICP	RGB	PBR+real	RGB-D	75.2	72.7	94.8	65.2	52.7	88.3	86.6	76.5	0.50
15	Extended_FCOS+PFA-PBR-RGBD [27]	2022	DNN	Dataset	Extended FCOS	PFA	RGB	PBR	RGB-D	79.7	80.2	89.3	67.6	46.9	86.9	82.6	76.2	2.63
16	SurfEmb-PBR-RGBD [15]	2022	DNN	Dataset	Default (PBR)	Custom	RGB-D	PBR	RGB-D	76.0	82.8	85.4	65.9	53.8	86.6	79.9	75.8	9.05
17	ZebraPoseSAT-EffnetB4 [53]	2023	DNN	Object	Default GDRNPPDet	–	RGB	PBR+real	RGB	72.9	82.1	85.0	59.2	50.4	92.2	82.8	74.9	2.50
18	GDRNPP-PBRReal-RGBD-SModel [39, 61]	2022	DNN	Dataset	YOLOX	Depth adjust.	RGB	PBR+real	RGB-D	75.7	85.6	90.6	68.0	35.6	86.4	81.7	74.8	0.56
19	Megapose-GDRNPPDet-PBRReal+Multi [33, 39]	2023	DNN	Dataset	Default GDRNPPDet	Teaser++	RGB	PBR	RGB-D	70.4	71.8	91.6	59.2	55.3	87.2	85.5	74.4	93.26
20	Coupled Iterative Refinement (CIR) [37]	2022	DNN	Dataset	Default (synt+real)	CIR	RGB-D	PBR+real	RGB-D	73.4	77.6	96.8	67.6	38.1	75.7	89.3	74.1	–
21	GPose2023-RGB [39, 61]	2023	DNN	Object	GDet2023	CIR	RGB	PBR	RGB	69.9	79.9	83.1	62.6	46.0	87.6	80.9	72.9	0.24
22	GDRNPP-PBRReal-RGB-MModel [39, 61]	2022	DNN	Object	YOLOX	–	RGB	PBR+real	RGB	71.3	78.6	83.1	62.3	44.8	86.9	82.5	72.8	0.23
23	ZebraPoseSAT-EffnetB4 [53]	2022	DNN	Object	FCOS	–	RGB	PBR+real	RGB	72.1	80.6	85.0	54.5	41.0	88.2	83.0	72.0	0.25
24	ZebraPoseSAT-EffnetB4(DefaultDet) [53]	2022	DNN	Object	Default (synt+real)	–	RGB	PBR+real	RGB	70.7	76.8	84.9	59.7	41.7	88.7	81.6	72.0	0.25
25	ZebraPoseSAT-EffnetB4(PBR-DefaultDet) [53]	2023	DNN	Object	Default GDRNPPDet	–	RGB	PBR+real	RGB	72.9	81.1	75.6	59.2	50.4	92.1	72.9	72.0	0.25
26	ZebraPose-SAT [53]	2022	DNN	Object	FCOS	–	RGB	PBR+real	RGB	72.1	78.7	86.1	54.9	37.9	84.7	82.8	71.0	–
27	Extended_FCOS+PFA-MixPBR-RGB [27]	2022	DNN	Dataset	Extended FCOS	PFA	RGB	PBR+real	RGB	74.5	77.8	83.9	60.0	35.3	84.1	80.6	70.9	3.02
28	GDRNPP-PBR-RGB-MModel [39, 61]	2022	DNN	Object	YOLOX	–	RGB	PBR	RGB	71.3	79.6	75.2	62.3	44.8	86.9	71.3	70.2	0.28
29	GDRNPPDet-PBRReal+GenFlow-Multi [39, 61]	2023	DNN	Dataset	Default GDRNPPDet	–	RGB	PBR+real	RGB	66.8	82.3	76.0	58.1	48.6	89.3	69.8	70.1	35.36
30	CosyPose-ECCV20-SYNT+REAL-ICP [32]	2020	DNN	Dataset	Default (synt+real)	DeepIM+ICP	RGB	PBR+real	RGB-D	71.4	70.1	93.9	64.7	31.3	71.2	86.1	69.8	13.74
31	MRPE-PBRReal-RGB-SModel	2023	DNN	Dataset	Default GDRNPPDet	Render & com.	RGB	PBR+real	RGB-D	74.4	75.8	82.4	55.0	36.8	77.0	84.3	69.4	0.10
32	GDRNPP-PBRReal-RGB-SModel	2022	DNN	Dataset	YOLOX	CIR	RGB	PBR+real	RGB	68.6	77.6	82.7	61.7	26.0	80.9	76.8	67.8	0.46
33	Megapose-GDRNPPDet-PBRReal+MultiHyp [33, 39]	2023	DNN	Dataset	Default GDRNPPDet	–	RGB	PBR	RGB	64.8	78.1	74.1	56.9	42.2	86.3	70.2	67.5	36.28
34	ZebraPoseSAT-EffnetB4 (PBR_Only) [53]	2022	DNN	Object	FCOS	–	RGB	PBR	RGB	72.1	72.3	71.7	54.5	41.0	88.2	69.1	67.0	–
35	Extended_FCOS+PFA-PBR-RGB [27]	2022	DNN	Dataset	Extended FCOS	PFA	RGB	PBR	RGB	74.5	71.9	73.2	60.0	35.3	84.1	64.8	66.3	3.50
36	PFA-cosypose [27, 32]	2022	DNN	Dataset	MaskRCNN	PFA	RGB-D	PBR+real	RGB	67.4	73.8	83.7	59.6	24.6	71.2	80.7	65.9	–
37	Megapose-GDRNPPDet_PBRReal [33]	2022	DNN	Dataset	Default GDRNPPDet	DeepIM	RGB	–	RGB	61.2	76.6	72.3	55.5	40.2	85.1	69.2	65.7	32.35
38	SurfEmb-PBR-RGB [15]	2022	DNN	Dataset	Default (PBR)	Custom	RGB	PBR	RGB	66.3	73.5	71.5	58.8	41.3	79.1	64.7	65.0	8.89
39	Koenig-Hybrid-DL-PointPairs [31]	2020	DNN/PPF	Dataset	Retina/MaskRCNN	ICP	RGB	Synt+real	RGB-D	63.1	65.5	92.0	43.0	48.3	65.1	70.1	63.9	0.63
40	CosyPose-ECCV20-SYNT+REAL-1VIEW [32]	2020	DNN	Dataset	Default (synt+real)	∼DeepIM	RGB	PBR+real	RGB	63.3	72.8	82.3	58.3	21.6	65.6	82.1	63.7	0.45
41	CRT-6D	2022	DNN	Dataset	Default (synt+real)	Custom	RGB	PBR+real	RGB	66.0	64.4	78.9	53.7	20.8	60.3	75.2	59.9	0.06
42	Pix2Pose-BOP20_w/ICP-ICCV19 [46]	2020	DNN	Object	MaskRCNN	ICP	RGB	PBR+real	RGB-D	58.8	51.2	82.0	39.0	35.1	69.5	78.0	59.1	4.84
43	ZTE_PPF	2022	DNN/PPF	Dataset	Default (synt+real)	ICP	RGB	PBR+real	RGB-D	66.3	37.4	90.4	39.6	47.0	73.5	50.2	57.8	0.90
44	CosyPose-ECCV20-PBR-1VIEW [32]	2020	DNN	Dataset	Default (PBR)	∼DeepIM	RGB	PBR	RGB	63.3	64.0	68.5	58.3	21.6	65.6	57.4	57.0	0.48
45	Vidal-Sensors18 [60]	2019	PPF	–	–	ICP	–	–	D	58.2	53.8	87.6	39.3	43.5	70.6	45.0	56.9	3.22
46	CDPNv2_BOP20 (RGB-only & ICP) [34]	2020	DNN	Object	FCOS	ICP	RGB	Synt+real	RGB-D	63.0	46.4	91.3	45.0	18.6	71.2	61.9	56.8	1.46
47	Drost-CVPR10-Edges [10]	2019	PPF	–	–	ICP	–	–	RGB-D	51.5	50.0	85.1	36.8	57.0	67.1	37.5	55.0	87.57
48	MRPE-PBR-RGB-SModel	2023	DNN	Dataset	Default GDRNPPDet	–	RGB	PBR	RGB	71.5	72.9	20.6	46.2	35.3	76.5	55.2	54.0	0.10
49	CDPNv2_BOP20 (PBR-only & ICP) [34]	2020	DNN	Object	FCOS	ICP	RGB	PBR	RGB-D	63.0	43.5	79.1	45.0	18.6	71.2	53.2	53.4	1.49
50	CDPNv2_BOP20 (RGB-only) [34]	2020	DNN	Object	FCOS	–	RGB	Synt+real	RGB	62.4	47.8	77.2	47.3	10.2	72.2	53.2	52.9	0.94
51	Drost-CVPR10-3D-Edges [10]	2019	PPF	–	–	ICP	–	–	D	46.9	40.4	85.2	37.3	46.2	62.3	31.6	50.0	80.06
52	Drost-CVPR10-3D-Only [10]	2019	PPF	–	–	ICP	–	–	D	52.7	44.4	77.5	38.8	31.6	61.5	34.4	48.7	7.70
53	CDPN_BOP19 (RGB-only) [34]	2020	DNN	Object	RetinaNet	–	RGB	Synt+real	RGB	56.9	49.0	76.9	32.7	6.7	67.2	45.7	47.9	0.48
54	CDPNv2_BOP20 (PBR-only & RGB-only) [34]	2020	DNN	Object	FCOS	–	RGB	PBR	RGB	62.4	40.7	58.8	47.3	10.2	72.2	39.0	47.2	0.98
55	leaping from 2D to 6D [38]	2020	DNN	Object	Unknown	–	RGB	Synt+real	RGB	52.5	40.3	75.1	34.2	7.7	65.8	54.3	47.1	0.43
56	EPOS-BOP20-PBR [20]	2020	DNN	Dataset	–	–	RGB	PBR	RGB	54.7	46.7	55.8	36.3	18.6	58.0	49.9	45.7	1.87
57	Drost-CVPR10-3D-Only-Faster [10]	2019	PPF	–	–	ICP	–	–	D	49.2	40.5	69.6	37.7	27.4	60.3	33.0	45.4	1.38
58	Félix&Neves-ICRA2017-IET2019 [48, 51]	2019	DNN/PPF	Dataset	MaskRCNN	ICP	RGB-D	Synt+real	RGB-D	39.4	21.2	85.1	32.3	6.9	52.9	51.0	41.2	55.78
59	Sundermeyer-IJCV19+ICP [56]	2019	DNN	Object	RetinaNet	ICP	RGB	Synt+real	RGB-D	23.7	48.7	61.4	28.1	15.8	50.6	50.5	39.8	0.86
60	Zhigang-CDPN-ICCV19 [34]	2019	DNN	Object	RetinaNet	–	RGB	Synt+real	RGB	37.4	12.4	75.7	25.7	7.0	47.0	42.2	35.3	0.51
61	PointVoteNet2 [12]	2020	DNN	Object	–	ICP	RGB-D	PBR	RGB-D	65.3	0.4	67.3	26.4	0.1	55.6	30.8	35.1	–
62	Pix2Pose-BOP20-ICCV19 [46]	2020	DNN	Object	MaskRCNN	–	RGB	PBR+real	RGB	36.3	34.4	42.0	22.6	13.4	44.6	45.7	34.2	1.22
63	Sundermeyer-IJCV19 [56]	2019	DNN	Object	RetinaNet	–	RGB	Synt+real	RGB	14.6	30.4	40.1	21.7	10.1	34.6	44.6	28.0	0.20
64	SingleMultiPathEncoder-CVPR20 [54]	2020	DNN	All	MaskRCNN	–	RGB	Synt+real	RGB	21.7	31.0	33.4	17.5	6.7	29.3	28.9	24.1	0.19
65	DPOD (synthetic) [66]	2019	DNN	Dataset	–	–	RGB	Synt	RGB	16.9	8.1	24.2	13.0	0.0	28.6	22.2	16.1	0.23

Table 3. **6D localization of seen objects (Task 1) on the seven core datasets.** The methods are ranked by the AR_C score which is the average of the per-dataset AR_D scores defined in Sec. 2.1. The last column shows the average image processing time in seconds, *i.e.*, the average time to localize all objects in an image (measured on different computers by the participants). Column *Year* is the year of submission, *Type* indicates whether the method relies on deep neural networks (DNN's) or point pair features (PPF's), *DNN per...* shows how many DNN models were trained, *Det./seg.* is the object detection or segmentation method, *Refinement* is the pose refinement method, *Train im.* and *Test im.* show image channels used at training and test time respectively, and *Train im. type* is the domain of training images. All test images are real.

#	Method	Year	Type	DNN per	Det./seg.	Refinement	Train im.	...type	Test im.	LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	AR_C	Time
1	GenFlow-MultiHypo16 [40]	2023	DNN	All	CNOS-fastSAM	Recurrent Flow	RGB-D	PBR	RGB-D	63.5	52.1	86.2	53.4	55.4	77.9	83.3	67.4	34.58
2	GenFlow-MultiHypo [40]	2023	DNN	All	CNOS-fastSAM	Recurrent Flow	RGB-D	PBR	RGB-D	62.2	50.9	84.9	52.4	54.4	77.0	81.8	66.2	21.46
3	Megapose-CNOS+Multih_Teaserpp-10 [33, 43]	2023	DNN	All	CNOS-fastSAM	MegaPose+Teaser++	RGB	PBR	RGB-D	62.6	48.7	85.1	46.7	46.8	73.0	76.4	62.8	141.97
4	Megapose-CNOS+Multih_Teaserpp-10 [33, 43]	2023	DNN	All	CNOS-fastSAM	MegaPose+Teaser++	RGB	PBR	RGB-D	62.0	48.5	84.6	46.2	46.0	72.5	76.4	62.3	116.56
5	SAM6D-CNOSmask [35, 43]	2023	DNN	All	CNOS-fastSAM	Cross-attention	RGB-D	PBR	RGB-D	64.8	48.3	79.4	50.4	35.1	72.7	80.4	61.6	3.87
6	PoZe (CNOS)	2023	DNN	All	CNOS-fastSAM	ICP	RGB-D	Custom	RGB-D	64.4	49.4	92.4	40.9	51.6	71.2	61.1	61.6	159.43
7	ZeroPose-Multi-Hypo-Refinement [3, 43]	2023	DNN	All	CNOS-fastSAM	MegaPose	RGB-D	PBR+real	RGB-D	53.8	40.0	83.5	39.2	52.1	65.3	65.3	57.0	16.17
8	GenFlow-MultiHypo-RGB	2023	DNN	All	CNOS-fastSAM	Recurrent Flow	RGB-D	PBR	RGB	56.3	52.3	68.4	45.3	39.5	73.9	63.3	57.0	20.89
9	Megapose-CNOS_fastSAM+Multih-10 [33, 43]	2023	DNN	All	CNOS-fastSAM	MegaPose	RGB	PBR	RGB	56.0	50.8	68.7	41.9	34.6	70.6	62.0	54.9	53.88
10	Megapose-CNOS_fastSAM+Multih [33, 43]	2023	DNN	All	CNOS-fastSAM	MegaPose	RGB	PBR	RGB	56.0	50.7	68.4	41.4	33.8	70.4	62.1	54.7	47.39
11	ZeroPose-Multi-Hypo-Refinement [3, 43]	2023	DNN	All	SAM + ImageBind	MegaPose	RGB-D	PBR+real	RGB-D	49.3	34.2	79.0	39.6	46.5	62.9	62.3	53.4	18.97
12	MegaPose-CNOS_fastSAM [33, 43]	2023	DNN	All	CNOS-fastSAM	MegaPose	RGB	PBR	RGB	49.9	47.7	65.3	36.7	31.5	65.4	60.1	50.9	31.72
13	ZeroPose-One-Hypo [3]	2023	DNN	All	SAM + ImageBind	MegaPose	RGB-D	PBR+real	RGB-D	27.2	15.6	53.6	30.7	36.2	46.2	34.1	34.8	9.76
14	GenFlow-coarse	2023	DNN	All	CNOS-fastSAM	–	RGB	PBR	RGB	25.0	21.5	30.0	16.8	15.4	28.3	27.7	23.5	3.84

Table 4. **6D localization of unseen objects (Task 4) on the seven core datasets.** The methods are ranked by the AR_C score which is the average of the per-dataset AR_D scores defined in Sec. 2.4. The last column shows the average image processing time (in seconds). Other columns as in Tab. 3.

#	Method	...based on	Year	Data	...type	AP_C	Time
1	GDet2023	YOLOv8	2023	RGB	PBR+real	79.8	.204
2	GDRNPPDet	YOLOX	2022	RGB	PBR+real	77.3	.081
3	GDet2023-PBR	YOLOv8	2023	RGB	PBR	76.9	.204
4	GDRNPPDet	YOLOX	2022	RGB	PBR	73.8	.081
5	Extended_FCOS	FCOS	2022	RGB	PBR+real	72.1	.030
6	Extended_FCOS	FCOS	2022	RGB	PBR	66.7	.030
7	DLZDet	DLZDet	2022	RGB	PBR	65.6	-
8	CosyPose	Mask R-CNN	2020	RGB	PBR+real	60.5	.054
9	CosyPose	Mask R-CNN	2020	RGB	PBR	55.7	.055
10	FCOS-CDPN	FCOS	2022	RGB	PBR	50.7	.047

Table 5. **2D detection of seen objects (Task 2).** The methods are ranked by the AP_C score defined in Sec. 2.2. The last column shows the average image processing time (in seconds).

#	Method	...based on	Year	Data	...type	AP_C	Time
1	ZebraPoseSAT	GDRNPP+Zebra	2023	RGB	PBR+real	61.9	.080
2	ZebraPoseSAT	GDRNPP+Zebra	2022	RGB	PBR+real	58.7	.080
3	ZebraPoseSAT	CDPNv2+Zebra	2023	RGB	PBR+real	57.9	.080
4	ZebraPoseSAT	CDPNv2+Zebra	2022	RGB	PBR+real	57.8	.080
5	ZebraPoseSAT	CosyPose+Zebra	2022	RGB	PBR	53.8	.080
6	ZebraPoseSAT	CDPNv2+Zebra	2022	RGB	PBR	52.3	.080
7	DLZDet	DLZDet	2022	RGB	PBR+real	49.6	-
8	DLZDet	DLZDet	2022	RGB	PBR	42.9	-
9	CosyPose	Mask R-CNN	2020	RGB	PBR+real	40.5	.054
10	CosyPose	Mask R-CNN	2020	RGB	PBR	36.2	.055

Table 6. **2D segmentation of seen objects (Task 3).** Details as in Tab. 5.

#	Method	Year	Train. im.	...type	Test. im.	AP_C	Time
1	CNOS_FastSAM [43]	2023	RGB	PBR	RGB	42.8	0.221
2	CNOS_SAM [43]	2023	RGB	PBR	RGB	36.1	1.847
3	ZeroPose [3]	2023	RGB	PBR	RGB	34.1	3.821

Table 7. **2D detection of unseen objects (Task 5).** The methods are ranked by the AP_C score defined in Sec. 2.5. The last column shows the average image processing time (in seconds).

#	Method	Year	Train. im.	...type	Test. im.	AP_C	Time
1	CNOS_FastSAM [43]	2023	RGB	PBR	RGB	41.2	0.221
2	CNOS_SAM [43]	2023	RGB	PBR	RGB	40.3	1.847
3	ZeroPose [3]	2023	RGB	PBR	RGB	37.2	3.821
4	lcc-fastsam	2023	RGB	PBR	RGB	14.9	1.182

Table 8. **2D segment. of unseen objects (Task 6).** Details as in Tab. 7.

4.4. Results on Task 4

The new task of 6D localization of unseen objects received 14 entries, as presented in Tab. 4. MegaPose [33], a method from 2022, was considered as the baseline and consists of two stages: (1) coarse object pose estimation by finding the rendered template image that is closest to the test image crop, and (2) pose refinement via a render-and-compare strategy. The RGB-only entry Megapose-CNOS_fastSAM+Multih-10 (#9) achieves 54.9 AR_C and further improves to 62.8 AR_C by using RGB-D images and an additional refinement with Teaser++ [65], see Megapose-CNOS+Multih_Teaserpp-10 (#3).

GenFlow-MultiHypo16 (#1), the best method for 6D localization of unseen objects, reaches 67.4 AR_C. This is a remarkable result since the performance is comparable to CosyPose [32], the best method in 6D localization of seen objects from 2020. GenFlow improves the coarse pose estimation stage of MegaPose by running the coarse network in a GMM-based hierarchical manner. For pose refinement, GenFlow adapts the recurrent flow network [13] to also estimate a visibility mask and replaces the pose regression network with a differentiable P*n*P solver.

Results in Tab. 4 highlight that the run time is a significant challenge for solving unseen object pose localization. While GenFlow-MultiHypo16 improved the run time by 4x compared to MegaPose, it still takes 34.58 s per image. SAM6D (#5) [35] based on GeoTransformer [47] is the fastest method by a significant margin with 3.87 s per image while still reaching 61.6 AR_C (-5.8 AR_C compared to Genflow-MultiHypo16 #1). Figure 3 shows qualitative comparison of the best method for unseen objects, GenFlow, with the best method for seen objects, GPose.

4.5. Results on Tasks 5 and 6

2D detection and segmentation of unseen objects in cluttered, occluded environments is a challenging task. Still, as shown in Tab. 7 and Tab. 8, the best method CNOS-FastSAM [43] reaches accuracy of 42.8 mAP_C in detection and 41.2 mAP in segmentation of unseen objects. For comparison, the instance segmentation accuracy is comparable to Mask R-CNN [16] that reached 40.5 mAP_C in the BOP challenge 2020 [25] while being trained on more than 1M synthetic and real images of the target objects. CNOS-FastSAM [43] instead relies on DINOv2 [45] features extracted from only 200 rendered reference views per object. All submitted detection and segmentation approaches are RGB-based and rely on SAM-like (Segment Anything) [35] methods to segment object instances in the image.

Despite the substantial progress in unseen object detection and segmentation driven by foundation models, there is still a relatively large gap to methods trained to detect and segment specific objects (compare Tab. 5 and 6). Especially, the amodal detection of occluded instances, *i.e.*, including occluded parts, is a clear challenge for approaches focusing on unseen objects, leading to a gap of 37 mAP_C between CNOS and GDet2023.

To what extent is this gap in 2D detection performance responsible for the gap in 6D localization of seen and unseen objects? When combined with the default GDRNPPDet detections from Task 2, the best method for 6D localization of unseen objects (GenFlow-MultiHypo16) achieves the pose accuracy of 79.2 AR_C (#10 Tab. 3). Since this is only 5.9 AR_C behind GDRNPPDet + GPose2023 (#2), we conclude that better methods for unseen object detection would provide great potential for improving methods for unseen object localization.

5. Awards

The BOP Challenge 2023 awards were presented at the 8th Workshop on Recovering 6D Object Pose[6] at the ICCV 2023 conference. The awards are based on the results analyzed in Sec. 4. The submissions were prepared by the following authors:

- GPose2023 and GDet2023 [67] by Ruida Zhang, Ziqin Huang, Gu Wang, Xingyu Liu, Chenyangguang Zhang, Xiangyang Ji
- GDRNPP [39, 61] by Xingyu Liu, Ruida Zhang, Chenyangguang Zhang, Bowen Fu, Jiwen Tang, Xiquan Liang, Jingyi Tang, Xiaotian Cheng, Yukang Zhang, Gu Wang, Xiangyang Ji
- OfficialDet-PFA [27] by Xinyao Fan, Fengda Hao, Yang Hai, Jiaojiao Li, Rui Song, Haixin Shi, Mathieu Salzmann, David Ferstl, Yinlin Hu

[6] cmp.felk.cvut.cz/sixd/workshop_2023

- ZebraPoseSAT [53] by Praveen Annamalai Nathan, Sandeep Prudhvi Krishna Inuganti, Yongliang Lin, Yongzhi Su,Yu Zhang, Didier Stricker, Jason Rambach
- Coupled Iterative Refinement [37] by Lahav Lipson, Zachary Teed, Ankit Goyal, Jia Deng
- GenFlow [40] by Sungphill Moon, Hyeontae Son.
- SAM6D [35] by Jiehong Lin, Lihua Liu, Dekun Lu, Kui Jia
- MegaPose [33] by Elliot Maitre, Mederic Fourmy, Lucas Manuelli, Yann Labbé
- PoZe by Andrea Caraffa, Davide Boscaini, Fabio Poiesi
- CNOS [43] by Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, Tomas Hodan

Awards for 6D localization of seen objects (Task 1):

- **The Overall Best Method:**
GPose2023
- **The Best RGB-Only Method:**
ZebraPoseSAT-EffnetB4
- **The Best Fast Method (less than 1s per image):**
GDRNPP-PBRReal-RGBD-MModel-Fast
- **The Best BlenderProc-Trained Method:**
GPose2023-PBR
- **The Best Single-Model Method** (trained per dataset)**:**
OfficialDet-PFA-Mixpbr-RGB-D
- **The Best Open-Source Method:**
GDRNPP-PBRReal-RGBD-MModel
- **The Best Method Using Default Detections:**
GPose2023-OfficialDet
- **The Best Method on T-LESS, ITODD, HB, IC-BIN:**
GPose2023
- **The Best Method on LM-O, YCB-V:**
GPose2023-OfficialDet
- **The Best Method on TUD-L:**
Coupled Iterative Refinement (CIR)

Awards for 2D detect./segment. of seen objects (Tasks 2 and 3):

- **The Overall Best Detection Method:**
GDet2023
- **The Best BlenderProc-Trained Detection Method:**
GDet2023-PBR
- **The Overall Best Segmentation Method:**
ZebraPoseSAT-EffnetB4 (DefaultDetection)
- **The Best BlenderProc-Trained Segment. Method:**
ZebraPoseSAT-EffnetB4 (DefaultDet+PBR_Only)

Awards for 6D localization of unseen objects (Task 4):

- **The Overall Best Method:**
GenFlow-MultiHypo16
- **The Best RGB-Only Method:**
GenFlow-MultiHypo-RGB
- **The Best Fast Method (less than 1s per image):**
SAM6D-CNOSmask
- **The Best BlenderProc-Trained Method:**
GenFlow-MultiHypo16
- **The Best Single-Model Method (one for all core datasets) :**
GenFlow-MultiHypo16
- **The Best Open-Source Method:**
Megapose-CNOS_fastSAM+Multih_Teaserpp-10
- **The Best Method Using Default Detections/Segmentations:**
GenFlow-MultiHypo16
- **The Best Method on ITODD, IC-BIN, HB, YCB-V:**
GenFlow-MultiHypo16
- **The Best Method on T-LESS:**
GenFlow-MultiHypo-RGB
- **The Best Method on LM-O:**
SAM6D-CNOSmask
- **The Best Method on TUD-L:**
PoZe (CNOS)

Awards for 2D detect./segment. of unseen objects (Tasks 5 and 6):

- **The Overall Best Detection Method:**
CNOS (FastSAM)
- **The Best BlenderProc-Trained Detection Method:**
CNOS (FastSAM)
- **The Overall Best Segmentation Method:**
CNOS (FastSAM)
- **The Best BlenderProc-Trained Segment. Method:**
CNOS (FastSAM)

6. Conclusions

Although the accuracy scores start saturating on the seen-object tasks (Tasks 1–3), the top-performing methods still need to improve efficiency in order to support real-time applications. 2023 was a strong first year for the new unseen-object tasks (Tasks 4–6), with the top performing method for 6D localization of unseen objects reaching the accuracy of the top 2020 method for 6D localization of seen objects. However, we identified a great potential in improving detection of occluded objects and making unseen object pose estimation more efficient. In 2023, methods for unseen objects were provided 3D mesh models to onboard the target objects. Next years, we are planning to introduce an even more challenging variant where only reference images of each object are provided for the onboarding. The evaluation system at bop.felk.cvut.cz stays open and raw results of all methods are publicly available.

References

[1] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D object pose estimation using 3D object coordinates. In *ECCV*, 2014. 4, 5

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 4

[3] Jianqiu Chen, Mingshan Sun, Tianpeng Bao, Rui Zhao, Liwei Wu, and Zhenyu He. 3d model-based zero-shot pose estimation pipeline. *arXiv preprint arXiv:2305.17934*, 2023. 6, 7

[4] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Dmitry Olefir, Tomáš Hodaň, Youssef Zidan, Mohamad

Elbadrawy, Markus Knauer, Harinandan Katam, and Ahsan Lodhi. BlenderProc: Reducing the reality gap with photorealistic rendering. *RSS Workshops*, 2020. 2, 4

[5] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019. 2, 4

[6] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Wendelin Knauer, Klaus H Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023. 4

[7] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malassiotis, and Tae-Kyun Kim. Recovering 6D object pose and predicting next-best-view in the crowd. In *CVPR*, 2016. 4

[8] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3D scanned household items. *ICRA*, 2022. 4

[9] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Hartinger, and Carsten Steger. Introducing MVTec ITODD – A dataset for 3D object recognition in industry. In *ICCVW*, 2017. 4

[10] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3D object recognition. *CVPR*, 2010. 1, 2, 6

[11] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding YOLO series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 5

[12] Frederik Hagelskjær and Anders Glent Buch. PointPoseNet: Accurate object detection and 6 DoF pose estimation in point clouds. *arXiv preprint arXiv:1912.09057*, 2019. 6

[13] Yang Hai, Rui Song, Jiaojiao Li, and Yinlin Hu. Shape-constraint recurrent flow for 6d object pose estimation. In *CVPR*, pages 4831–4840, 2023. 7

[14] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, light & material decomposition from images using monte carlo rendering and denoising. *NeurIPS*, 2022. 1

[15] Rasmus Laurvig Haugaard and Anders Glent Buch. SurfEmb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. *CVPR*, 2022. 6

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *ICCV*, 2017. 2, 7

[17] 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. *ACCV*, 2012. 4

[18] Stefan Hinterstoisser, Vincent Lepetit, Paul Wohlhart, and Kurt Konolige. On pre-trained image features and synthetic images for deep learning. *ECCVW*, 2018. 2

[19] Tomáš Hodaň. Pose estimation of specific rigid objects. *PhD Thesis, Czech Technical University in Prague*, 2021. 4

[20] Tomáš Hodaň, Dániel Baráth, and Jiří Matas. EPOS: Estimating 6D pose of objects with symmetries. *CVPR*, 2020. 6

[21] Tomáš Hodaň, Eric Brachmann, Bertram Drost, Frank Michel, Martin Sundermeyer, Jiří Matas, and Carsten Rother. BOP Challenge 2019. https://bop.felk.cvut.cz/media/bop_challenge_2019_results.pdf, 2019. 1

[22] Tomáš Hodaň, Pavel Haluza, Štěpán Obdržálek, Jiří Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. *WACV*, 2017. 1, 4

[23] Tomáš Hodaň, Jiří Matas, and Štěpán Obdržálek. On evaluation of 6D object pose estimation. *ECCVW*, 2016. 2

[24] Tomáš Hodaň, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, Caner Sahin, Fabian Manhardt, Federico Tombari, Tae-Kyun Kim, Jiří Matas, and Carsten Rother. BOP: Benchmark for 6D object pose estimation. *ECCV*, 2018. 1, 4

[25] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. BOP Challenge 2020 on 6D object localization. In *ECCV*, 2020. 1, 2, 3, 4, 5, 7

[26] Tomáš Hodaň, Vibhav Vineet, Ran Gal, Emanuel Shalev, Jon Hanzelka, Treb Connell, Pedro Urbina, Sudipta Sinha, and Brian Guenter. Photorealistic image synthesis for object instance detection. *ICIP*, 2019. 2

[27] Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Perspective flow aggregation for data-limited 6d object pose estimation. *arXiv preprint arXiv:2203.09836*, 2022. 6, 7

[28] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, Jan. 2023. 5

[29] Roman Kaskman, Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. HomebrewedDB: RGB-D dataset for 6D pose estimation of 3D objects. *ICCVW*, 2019. 4

[30] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. *ICCV*, 2017. 2

[31] Rebecca Koenig and Bertram Drost. A hybrid approach for 6dof pose estimation. *ECCVW*, 2020. 6

[32] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent multi-view multi-object 6D pose estimation. *ECCV*, 2020. 1, 2, 6, 7

[33] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare. In *CoRL*, 2022. 2, 4, 6, 7, 8

[34] Zhigang Li, Gu Wang, and Xiangyang Ji. CDPN: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. *ICCV*, 2019. 6

[35] Jiehong Lin, Lihua Liu, Dekun Lu, and Kui Jia. Sam-6d: Segment anything model meets zero-shot 6d object pose estimation. In *CVPR*, 2024. 6, 7, 8

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. *ECCV*, 2014. 3

[37] Lahav Lipson, Zachary Teed, Ankit Goyal, and Jia Deng. Coupled iterative refinement for 6d multi-object pose estimation. In *CVPR*, 2022. 6, 8

[38] Jinhui Liu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, Errui Ding, Feng Xu, and Xin Yu. Leaping from 2D detection to efficient 6DoF object pose estimation. *ECCVW*, 2020. 6

[39] Xingyu Liu, Ruida Zhang, Chenyangguang Zhang, Bowen Fu, Jiwen Tang, Xiquan Liang, Jingyi Tang, Xiaotian Cheng, Yukang Zhang, Gu Wang, and Xiangyang Ji. GDRNPP. https://github.com/shanice-l/gdrnpp_bop2022, 2022. 1, 2, 5, 6, 7

[40] Sungphill Moon, Hyeontae Son, Dongcheol Hur, and Sangwook Kim. GenFlow: Generalizable Recurrent Flow for 6D Pose Refinement of Novel Objects. In *arXiv preprint arXiv:2403.11510*, 2024. 1, 2, 5, 6, 8

[41] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. *CVPR*, 2022. 1

[42] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. *ISMAR*, 2011. 1, 4

[43] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. CNOS: A Strong Baseline for CAD-based Novel Object Segmentation. In *ICCVW*, 2023. 2, 3, 6, 7, 8

[44] Van Nguyen Nguyen, Yinlin Hu, Yang Xiao, Mathieu Salzmann, and Vincent Lepetit. Templates for 3D Object Pose Estimation Revisited: Generalization to New Objects and Robustness to Occlusions. In *CVPR*, 2022. 2

[45] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 7

[46] Kiru Park, Timothy Patten, and Markus Vincze. Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation. *ICCV*, 2019. 6

[47] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, Slobodan Ilic, Dewen Hu, and Kai Xu. Geotransformer: Fast and robust point cloud registration with geometric transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 7

[48] Carolina Raposo and Joao P Barreto. Using 2 point+normal sets for fast registration of point clouds with small overlap. *ICRA*, 2017. 6

[49] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *ICCV*, 2021. 1

[50] Colin Rennie, Rahul Shome, Kostas E Bekris, and Alberto F De Souza. A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place. *RA-L*, 2016. 4

[51] Pedro Rodrigues, Michel Antunes, Carolina Raposo, Pedro Marques, Fernando Fonseca, and Joao Barreto. Deep segmentation leverages geometric pose estimation in computer-aided total knee arthroplasty. *Healthcare Technology Letters*, 2019. 6

[52] Ivan Shugurov, Fu Li, Benjamin Busam, and Slobodan Ilic. Osop: A multi-stage one shot object pose estimation framework. In *CVPR*, 2022. 2

[53] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. ZebraPose: Coarse to fine surface encoding for 6DoF object pose estimation. *CVPR*, 2022. 5, 6, 8

[54] 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. *CVPR*, 2020. 6

[55] Martin Sundermeyer, Tomas Hodan, Yann Labbé, Gu Wang, Eric Brachmann, Bertram Drost, Carsten Rother, and Jiri Matas. BOP challenge 2022 on detection, segmentation and pose estimation of specific rigid objects. *CVPRW*, 2023. 1, 2

[56] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, and Rudolph Triebel. Augmented Autoencoders: Implicit 3D orientation learning for 6D object detection. *IJCV*, 2019. 6

[57] Alykhan Tejani, Danhang Tang, Rigas Kouskouridas, and Tae-Kyun Kim. Latent-class hough forests for 3D object detection and pose estimation. *ECCV*, 2014. 4

[58] Stephen Tyree, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Jeffrey Smith, and Stan Birchfield. 6-DoF pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark. *IROS*, 2022. 4

[59] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In *CVPR*, 2022. 1

[60] 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*, 2018. 1, 6

[61] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation. *CVPR*, 2021. 1, 2, 5, 6, 7

[62] Bojian Wu, Yang Zhou, Yiming Qian, Minglun Cong, and Hui Huang. Full 3D reconstruction of transparent objects. *ACM TOG*, 2018. 1

[63] Yangzheng Wu, Alireza Javaheri, Mohsen Zand, and Michael Greenspan. Keypoint cascade voting for point cloud based 6DoF pose estimation. *arXiv preprint arXiv:2210.08123*, 2022. 6

[64] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. *RSS*, 2018. 3, 4, 5

[65] H. Yang, J. Shi, and L. Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Trans. Robotics*, 2020. 7

[66] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. DPOD: 6D pose object detector and refiner. *ICCV*, 2019. 6

[67] Ruida Zhang, Ziqin Huang, Gu Wang, Xingyu Liu, Chenyangguang Zhang, and Xiangyang Ji. GPose2023, a submission to the BOP Challenge 2023. *Unpublished*, 2023. `http://bop.felk.cvut.cz/method_info/410/`. 1, 2, 5, 7