OSSID: Online Self-Supervised Instance Detection by (and for) Pose Estimation

Qiao Gu, Brian Okorn, and David Held

***Abstract*— Real-time object pose estimation is necessary for many robot manipulation algorithms. However, state-of-the-art methods for object pose estimation are trained for a specific set of objects; these methods thus need to be retrained to estimate the pose of each new object, often requiring tens of GPU-days of training for optimal performance. In this paper, we propose the OSSID framework, leveraging a slow zero-shot pose estimator to self-supervise the training of a fast detection algorithm. This fast detector can then be used to filter the input to the pose estimator, drastically improving its inference speed. We show that this self-supervised training exceeds the performance of existing zero-shot detection methods on two widely used object pose estimation and detection datasets, without requiring any human annotations. Further, we show that the resulting method for pose estimation has a significantly faster inference speed, due to the ability to filter out large parts of the image. Thus, our method for self-supervised online learning of a detector (trained using pseudo-labels from a slow pose estimator) leads to accurate pose estimation at real-time speeds, without requiring human annotations. Supplementary materials and code can be found at https://georgegu1997.github.io/OSSID/**

I. Introduction

Object instance detection and pose estimation are crucial to many robot manipulation tasks. Unlike the standard computer vision tasks of detecting all instances of a given semantic object category (such as person, car, or bicycle), for robotic manipulation, robots need to detect specific object instances. For example, a robot agent in the kitchen needs to distinguish a salt can from a coffee can, even though both objects may fall into the semantic category of "can."

In the past few years, deep convolutional neural networks have become the prevailing tool for object instance detection and pose estimation, outperforming alternative methodologies in various benchmarks [1], [2], [3], [4], [5]. However, most deep object detection and pose estimation methods are object-specific, which means the object categories are pre-defined and "baked into" the network weights. In other words, when we want to apply the detector and pose estimator to new objects, new data needs to be collected and the network needs to be retrained or finetuned on those new object instances. Current state-of-the-art methods for object pose estimation [4] require tens of GPU-days of training, which is quite a long time for a robot to wait every time a human selects a new brand of coffee. This quickly becomes infeasible as every new can, box, and tool requires us to repeat this process.

Q. Gu is with Department of Computer Science, University of Toronto. (qgu@cs.toronto.edu)

B. Okorn and D. Held are with the Robotics Institute, Carnegie Mellon University. (bokorn@andrew.cmu.edu, dheld@andrew.cmu.edu)

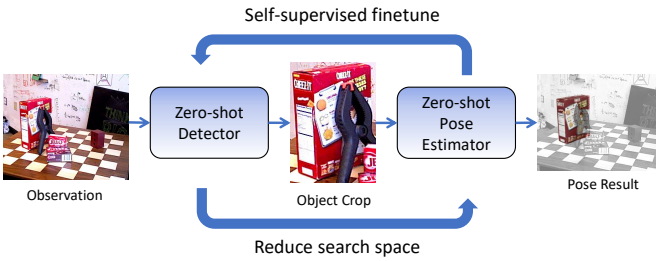


Fig. 1. We propose OSSID, a self-supervised learning pipeline for object instance detection by pose estimation. The results of a zero-shot pose estimation network are used to finetune a zero-shot detector online. Then the detection results in turn provide object bounding boxes and reduce the search space for pose estimation. Without any manual annotation required, both the detector and the pose estimator get better and faster.

To address this issue, a number of zero-shot pose estimators have been developed. However, most zero-shot pose estimators only evaluate on sparse, uncluttered scenes where the object of interest is detected and cropped or is sitting on an empty table [6], [7], [8]. Evaluation of such methods in cluttered settings shows that such methods fail to provide reasonable performance, even with the addition of ground-truth bounding boxes or ground truth translation as input (see analysis in Okorn, *et al.* [9] Appendix B). A recent method has directly tackled the challenge of zero-shot object pose estimation in clutter [9], but this approach has a very slow inference speed of 3 seconds per frame, due to the need to generate and evaluate pose hypotheses over a large 6D pose search space. This inference time is much too slow for real-time robotics applications.

In this work, we explore how a zero-shot object detector can be combined with a zero-shot pose estimator for faster performance, without loss of accuracy. Specifically, we build upon the work of Okorn *et al.* [9], but significantly increase the inference speed by using a zero-shot object detector to reduce the search space. This focuses the pose estimation on the smaller region of the image within a detected bounding box, instead of processing the entire image.

Unfortunately, a naive implementation of this straightforward combination does not give satisfactory performance. As our analysis shows, current zero-shot instance detectors only have mediocre performance when evaluated on objects outside their training set. Therefore, we propose to adapt the detector to novel objects and unseen environments with a zero-shot pose estimator.

We exploit the insight that a slow method for zero-shot pose estimation provides free and high-quality pseudo-

ground truth for training a fast object detector. As outlined in Fig. 1, we propose OSSID, an online self-supervised instance detection framework, using a zero-shot pose estimation pipeline to generate pseudo-ground truth detection and segmentation labels on the test environment. After performing self-supervised online learning, the resulting method for object instance detection and pose estimation outperforms baselines on both speed and accuracy by a large margin. In this work, we assume that the 3D mesh model of the target object is available; 3D object mesh models can be easily obtained using 3D reconstruction software [10], [11], [12], [13], [14], with an overhead of only several minutes.

We evaluate OSSID on two popular and challenging datasets: LineMOD-Occlusion and YCB-Video. The results demonstrate that online self-supervised learning using a zero-shot pose estimator can help a detector quickly adapt to new objects and new environments. Further, the detector reduces the search space for 6D poses and drastically improves the inference speed for pose estimation.

II. Related Work

A. Zero-shot Pose Estimation

Classical methods based on hand-crafted features perform no learning and thus are inherently zero-shot to different object instances. In the field of 6D pose estimation, Point Pair Features (PPF) and its variants can still achieve good results on recent benchmarks [15], [16], [17], [18], [19], [5]. However, PPF-based methods have much slower inference speed than deep-learned methods (typically an order of magnitude slower) and no longer match the accuracy of recent deep-learned methods.

Although deep learning methods for 6D pose estimation have achieved very accurate results, most such methods are trained for particular objects and do not generalize to unseen objects without retraining, which can take tens of GPU-days [20], [2], [21], [4]. Several recent works tackled the zero-shot pose estimation problem by learning a latent object representation [6], [7], [8]. However, recent analysis has revealed that such methods perform poorly in cluttered scenes, even when a ground-truth bounding box is provided as input [9]. In recent work, ZePHyR [9] overcomes this issue by learning to score many pose hypotheses in cluttered scenes. However, ZePHyR requires scoring a large number of pose hypotheses over the entire image. This results in a slow inference speed of up to 3 seconds per image per object, which prevents its use for real-time robotics applications. We propose to use self-supervised online learning to train a zero-shot instance detector to filter the input to ZePHyR, significantly speeding up its performance, without requiring any human annotations.

Related to our work, previous work has shown that the accuracy and inference speed of PPF can be improved when augmented with a deep learned object instance detection algorithm [22]. This work, however, required the training of an object-specific pose estimator, which itself requires large training times. In contrast, our approach can be trained quickly with online self-supervised learning.

B. Few-shot Object Detection

Much effort has been devoted to zero-shot or few-shot object detection by the computer vision community in the past few years [23], [24], [25], [26], [27], but most of these works have been focused on class-level semantic object detection. However, in the context of robot manipulation, robot agents often need to locate a specific object instance in a cluttered environment. Traditional methods tackled this problem using hand-crafted features and template matching [28], [29], [30], [31]. Recently, several deep learned methods have been proposed for few-shot object instance detection [32], [33]. While we build on the state-of-the-art zero-shot instance detector, DTOID [33], we demonstrate that this network only achieves mediocre performance on unseen object instances. We then show that our self-supervised learning pipeline can significantly improve the detection performance. We additionally show we can achieve a faster inference speed with a reduced number of object templates, while maintaining equivalent or better detection accuracy.

C. Domain Adaptation for Object Detection

Although deep CNNs have achieved significant progress on object detection, their performance will degrade on images out of the training distribution. Recent efforts have attempted to tackle this challenge by unsupervised domain adaptation [34], [35], [36], [37], [38]. We refer readers to Oza, *et al.* [39] for a comprehensive survey on unsupervised domain adaptation of object detection. Our method is similar to those based on pseudo-label based self-training [40], [41], [42], but we focus on object instance detection (rather than class level detection) and we obtain pseudo labels from zero-shot pose estimation. In the related area of domain adaptation for object tracking, Pirk *et al.* [43] uses a contrastive loss to learn object representations, showing that tracking performance increases with gradual online training. More closely related to our work, Mitash *et al.* [44] proposed a self-supervised online learning system for object detection using physics simulation and multi-view pose estimation. However, this method relies on large synthetic datasets, and their system does not generalize to objects that do not exist in the synthetic training set. In contrast, our method is able to adapt to unseen objects, improving the initial zero-shot detector as more scenes of the test environment are processed. Mitash *et al.* also assume the environment to be a clean tabletop or predefined shelves, limiting their range of application. In contrast, our method has been shown to work in cluttered environments, and is able to adapt to objects and environments outside of those it was initially trained in.

III. Method

Our goal is to train a fast and accurate pose estimator without requiring any human annotations or long training times. To achieve this, we proposed OSSID: online self-supervised instance detection, using a slow zero-shot pose estimator (ZePHyR [9]) to train a fast object instance detector through online self-supervision. This object instance detector can then be used to filter the input space of our pose

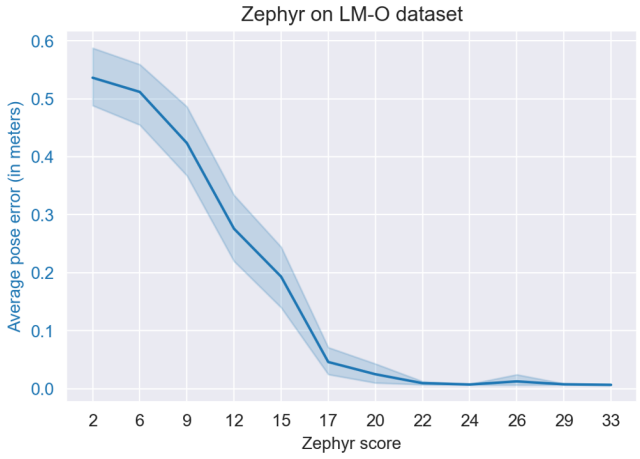


Fig. 2. Distribution of mean pose accuracy vs. Zephyr score of the top-scored pose hypothesis for all images in LM-O dataset, with 2 Std Dev shown in light blue. The plot demonstrates that ZePHyR returns higher scores for pose hypotheses with lower error. We use pose results with scores higher than 20 for self-supervised learning.

estimator, increasing the inference speed without reducing the accuracy of the overall system.

A. Zero-shot Pose Estimation

Recent work has shown that combining non-learned pose hypothesis generation with a deep learned fitness function can produce highly accurate pose estimates on objects never seen at training time [9]. This method, while having the ability to generalize to arbitrary objects with no retraining required, requires a long inference time to generate potential hypotheses across the full image space.

For object pose estimation, we adopt ZePHyR [9], a zero-shot pose scoring algorithm that generalizes to unseen objects without needing extra labeling or re-training. Following [9], we use Point Pair Features [15] and SIFT feature matching [45] for 6D object pose hypothesis generation.

The runtime of this pose estimation algorithm is strongly correlated to the number of pose hypotheses being evaluated. Therefore, we propose to use an object detector to filter the pose search space, removing unlikely regions of the input. Specifically, we crop the input scene using a learned object instance detector, generating hypotheses using only points from within the cropped region. As such, we do not generate hypotheses outside of this bounding box, which reduces the number of hypotheses that ZePHyR will evaluate, which reduces the inference time. Also, we do not generate features for the region outside of the detector bounding box, which further reduces the runtime. The combination of these benefits leads to a significant increase in inference speed, as described in Sec. IV-F.

B. Zero-shot Detection

As explained above, we aim to use a detector to filter pose hypotheses for ZePHyR, increasing the inference speed of the pose estimation pipeline. However, most object detectors are trained on a large dataset of the target objects, which requires waiting for many GPU-days for training to complete. In contrast, we hope to obtain a pose estimation system that can work quickly on a novel object, which requires that the detection system that will be used to filter hypotheses must be trained quickly as well.

Recently, researchers have proposed zero-shot methods for object instance detection (such as DTOID [33]). These networks are specially designed to compare an object template with the observation of a scene to find the target object. DTOID is trained on a large set of objects in an attempt to generalize this comparison to new objects without extra training. Specifically, for DTOID to detect a new object, it only requires template images of the target object. These template images can be generated by rendering images of an object mesh model [33]. We assume such object meshes are available, since they are provided for us in the datasets that we use for evaluation. For creating such mesh models, there are many techniques for 3D object reconstruction [10], [11], [12], [13], [14]; furthermore, with the help of 3D capturing software, one can capture a 3D mesh model of the object with a cellphone within several minutes [46], [47], [48].

However, we find that DTOID only has mediocre performance when tested on objects outside of the training distribution, as we show in our experiments in Sec. IV-D. We hypothesize that this is because of the large domain gap between the objects and environments during training and testing. Without adaptation to the unseen testing domain, the network generalizes poorly to out-of-distribution test examples.

To overcome this limitation, we propose a method for online self-supervised finetuning for object instance detection. Specifically, we evaluate a zero-shot pose estimator on previous frames of the target environment. We then use these pose estimates as pseudo-labels for self-supervision. Given these pseudo-labels, we can finetune an object instance detector, improving its performance on the target environment. The integration of this self-supervised detector will then improve the speed of the overall system by filtering hypotheses for the pose estimator.

C. Online Self-supervised Learning

We introduce OSSID, a self-supervised learning framework for online adaptation of an object detector to novel objects in an unseen testing environment, as depicted in Fig. 3. The key insight is that a 6D pose estimator predicts the full state of a rigid object; a zero-shot pose estimator can thus provide free supervision for training an object detector. As the pose estimator is only used for training, even a slow pose estimation method can be used, as speed is less relevant at training time. Once self-supervised finetuning is complete, the object detector provides a target region for the pose estimator and thus improves both the speed and accuracy of the pose estimator.

We assume access to a stream of RGB-D images I_t, each containing a target object with its mesh model M and the template images T; we process these images in sequential order, following the online learning paradigm [43]. Note

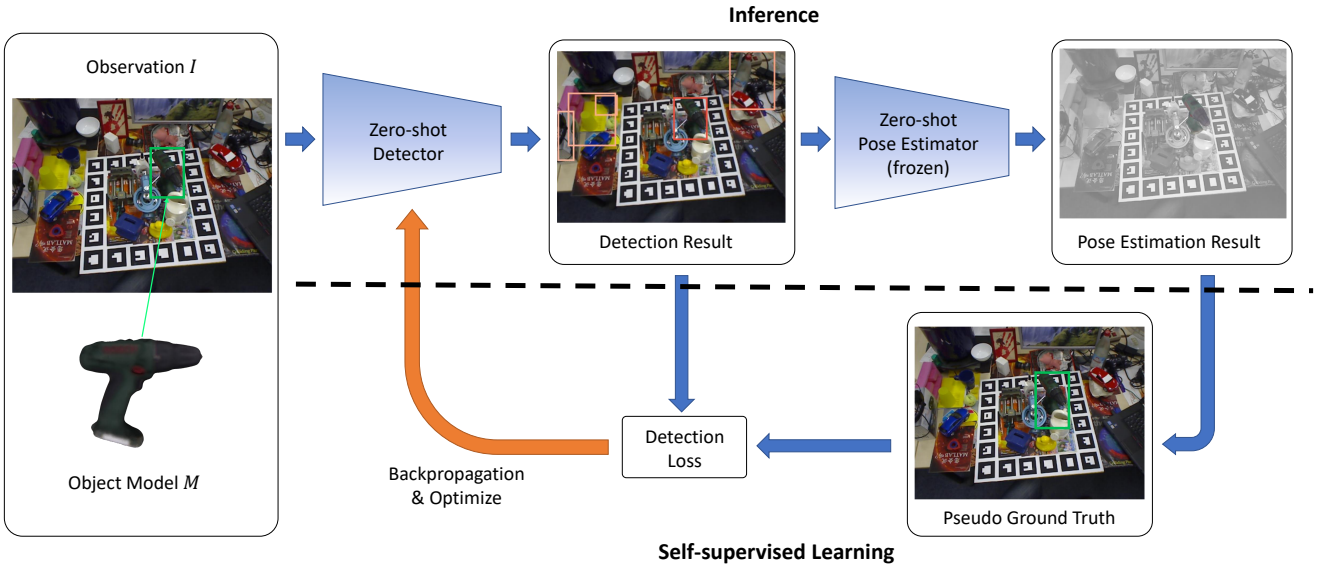


Fig. 3. Pipeline for OSSID: online self-supervised instance detection by (and for) 6D pose estimation. The upper part shows the pipeline for inference, where object detection results are used to filter the input of the pose estimator, increasing its inference speed. The lower part shows the self-supervised learning procedure. High scoring poses generated by the zero-shot pose estimator are used as pseudo ground truth to self-supervise the detection network, which helps the detector adapt and achieve better performance without any manual labels. The weights of the zero-shot pose estimator are not changed in this process.

that in this streaming pipeline, each image is seen only once. The detection and pose estimation results are used for both evaluation and online self-supervised learning. This aligns with the situation where a robot agent enters a new environment, encounters a new object, and gradually adapts its detector to find this object.

To obtain pseudo-ground truth labels for training the detector, we run the zero-shot pose estimation network, ZePHyR, on the observation image I. The output of ZePHyR is a pose estimation result $\{h, s\}$, where h is a pose and s is the score of the associated hypothesis. The question is: when should we trust a pose hypothesis output by the zero-shot pose estimation network? We ideally only want to train the object detector on pose hypotheses that are accurate, which we can treat as pseudo-ground truth.

Fortunately, we found that the score s generated by ZePHyR is a strong indicator of the accuracy of the pose hypothesis h. We visualize this in Fig. 2, which clearly demonstrates that, as the ZePHyR score increases, the error of the estimated pose goes down. As the figure shows, pose estimates with ZePHyR score greater than 20 are extremely accurate, with an average pose error of less than 2 centimeters. We can thus filter out pose hypotheses with a score less than 20; we treat pose hypotheses with a score of at least 20 as pseudo-ground truth for training an object detector. Note that we only consider the highest-scoring pose hypothesis in a given image, ignoring the possibility of training from multiple potential instances of the same object in a scene.

Thus, if the score s is larger than the threshold, we treat this pose as pseudo-ground truth and push it into a finetuning dataset F together with the observation I. This finetuning dataset is then used to finetune the detection network through backpropagation, as shown in Figure 3 ("Backpropagation and optimize"). Note that the input object templates (the rendered images) are not changed during training.

Note that 6D pose results can be easily converted to detection bounding boxes and segmentation masks by projecting the object model into the image frame. Thus, a pose estimate provides full supervision for training the detection network (bounding boxes or segmentation masks) without any human annotations.

To speed up the training process, rather than running the pose estimator on the entire image, we instead run the detector on the image I. We then only run the pose estimation method on a cropped region within the highest-scoring bounding box. We perform such cropping during both online self-supervised training of the detector (to speed up training) as well as during inference.

The full online self-supervised learning process is further detailed in Algorithm 1. Note that in this pipeline, the object detector is trained on the test dataset, but no manual annotations are used. As we can see from Sec. IV-G, the detection accuracies improve as the online self-supervised learning proceeds.

IV. Experiments

A. Dataset

For our experiments, we evaluate over two popular datasets, LM-O and YCB-V, which contain rigid objects with 6D pose annotations. These two datasets are challenging for object detection and pose estimation due to the presence of clutter, occlusions, and lighting variations. In our experiments, we assume the object mesh models are available,

Algorithm 1: OSSID: Online Self-supervised Instance Detection

input : A testing dataset D containing images I, a target object mesh M with template images T, a detector `Detect`, a pose estimator `Pose`, score threshold for good pose θ^p

output: A fine-tuned detector `Detect`

```
F ← {} ; //Dataset for online learning
foreach (I, M) in D do
    d ← Detect(I, T);
    I' ← Crop(I, d);
    {h, s} ← Pose(I', M);
    if s > θ^p then
        Append(F, (I, h));
    end
    Finetune(Detect, F);
end
return Detect
```

but we do not need any manual annotation or synthetic data generation for self-supervised learning.

LineMOD-Occlusion dataset (LM-O) [20] contains a single scene from the testing set of the larger LineMOD (LM) dataset [49]. While LM only provides 6D pose annotation for one object in each scene, LM-O densely annotates all 8 low-textured objects in the selected scene. For zero-shot pose estimation, we adopted the ZePHyR model [9] trained on a synthetic dataset containing LM objects that are not in the LM-O dataset [1]. For detection, we used the DTOID model weights provided by [33], which were also trained on a synthetic dataset containing various objects from BOP datasets, excluding those in LM and LM-O. In this way, both the detector and pose estimator did not see the LM-O objects during training and their results are zero-shot.

YCB-Video dataset (YCB-V) [2] includes 92 RGB-D videos captured with 21 YCB objects [50], densely annotated with detection bounding boxes, segmentation masks and 6D poses. This dataset is challenging for pose estimation as the videos have different lighting conditions, occlusions, and sensor noise. We evaluate our method on the BOP testing set [5], which is a subset of the 12 testing videos originally defined in [2] and contains testing images with higher-quality ground truth poses. For zero-shot pose estimation, we followed the testing protocol in Okorn *et al.* [9], adopting two models trained on complementary object sets and testing them on the objects that were not seen during training. For detection, since the model weights provided by Mercier *et al.* [33] use the YCB objects during training, we trained our own DTOID weights by creating a synthetic dataset without objects in YCB-V. Although this retrained version of DTOID has poor detection results on the YCB-V dataset at first (detection mAP of 11.6 in Table I), we observed a large improvement in detection after online self-supervised learning, even surpassing the non-zero-shot baseline using the weights provided by Mercier *et al.* [33] (Detection mAP of 63.7).

B. Metrics

We evaluate the performance of the detector and the pose estimator before and after self-supervised learning. To evaluate the detector, following previous work [33], we report the detection mean average precision (mAP) [51] using an IoU threshold of 0.5. For pose estimation accuracy, we follow the BOP Challenge [5] and report the average recall scores (AR). AR is the average of three pose accuracy metrics: Visible Surface Discrepancy (VSD), Maximum Symmetry-Aware Surface Distance (MSSD), and Maximum Symmetry-Aware Projection Distance (MSPD). For detailed formulations, we refer readers to Hodaň *et al.* [5].

C. Baselines

While there are many methods for object detection and pose estimation, most of them require large training datasets and do not generalize to unseen object instances. In contrast, our goal is to design a system for detection and pose estimation that can be trained very quickly without large training datasets. Therefore, we select DTOID [33] and ZePHyR [9] as the baseline methods for object detection and pose estimation respectively. DTOID [33] achieves the state-of-the-art results on zero-shot object instance detection, outperforming all other comparable approaches on the LM-O dataset [52], [32], [53]. Similarly, ZePHyR [9] is the state-of-the-art in zero-shot pose estimation on LM-O and YCB-V.

D. Online Self-supervised Learning

We compare the performance of the detection and pose estimation results of our method to zero-shot baselines in Table I. "OSSID (Ours)" shows the performance of our method when the instance detector is self-supervised finetuned online following Algorithm 1. As can be seen in this table, our method outperforms both DTOID and ZePHyR on both the detection mAP and pose AR metrics for both the LM-O and YCB-V datasets. Importantly, for pose estimation, we reduce the inference time compared to Zephyr by a factor of 14 on LM-O and a factor of almost 2 on YCB-V.

In our experiments, the DTOID network is optimized with a fixed set of object templates rendered from the object mesh model. We also experimented with using only 10 templates instead of 160 (though in each experiment, the number of templates is still held fixed throughout training). We report the difference in Table II. We found that, for our method where the network is learning online, reducing the number of templates has little impact on the performance of the detection network, but greatly increased the inference speed. Specifically, we can see that the detector trained using our method only needs 10 local templates to have comparable performance while achieving a real-time speed. In contrast, the original DTOID network has a large drop in performance on the segmentation mean IOU metric when the number of templates is reduced.

We also tested a variant of our method, in which we use an approach that we call "confidence filtering", reported

Dataset	Task	DTOID	ZePHyR	OSSID (Ours)	OSSID (w/ Conf. Filter)	OSSID (Oracle)	OSSID (Transductive)	CosyPose [4]
LM-O	Detection mAP	51.3	67.1	64.0	**67.3**	74.4	78.4	90.5
	Pose AR score	–	59.8	61.7	**63.9**	66.8	66.0	63.3
	Inference Time (ms)	430	2949	**210**	710	210	210	69
YCB-V	Detection mAP	11.6*	58.1	63.2	**64.0**	79.2	68.9	86.1
	Pose AR Score	–	51.6	53.3	**55.3**	55.3	57.1	57.4
	Inference Time (ms)	430	619	**320**	350	320	320	69

TABLE I

ZERO-SHOT DETECTION AND POSE ESTIMATION RESULTS. FOR OSSID, THE INFERENCE TIME REPORTED IS THE TOTAL TIME OF RUNNING BOTH THE DETECTOR AND THE POSE ESTIMATOR, AND THE DETECTION STAGE IN OUR METHODS ONLY TAKES 50 MS. WE ALSO REPORT THE DETECTION MAP OF BASELINE ZEPHYR BY CONVERTING ITS POSE ESTIMATION RESULTS TO BOUNDING BOXES. *NOTE THAT FOR THE DTOID BASELINE ON YCB-V, WE USE THE WEIGHTS TRAINED ON OUR OWN DATASET, SINCE THE PRE-TRAINED MODEL WAS TRAINED ON YCB-V OVER A LONG TRAINING PERIOD (WE ONLY COMPARE TO METHODS WITH SHORT, OR NON-EXISTENT (ZERO-SHOT), TRAINING TIMES).

Method	Number of local templates	Detection Time (ms)	Detection mAP	Segmentation mean IoU
DTOID	10	**50**	50.5	33.4
	160	430	51.3	41.6
OSSID	10	**50**	**64.0**	46.8
	160	430	62.7	**48.4**

TABLE II

EFFECT OF LOCAL TEMPLATES ON DTOID DETECTION NETWORK ON LM-O DATASET, WITH AND WITHOUT OUR SELF-SUPERVISED LEARNING PIPELINE.

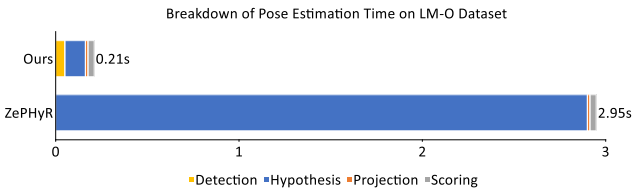


Fig. 4. Breakdown of the inference time of pose estimation using ZePHyR [9] on the LM-O dataset. Note that the time for pose hypothesis generation is greatly reduced on detected regions for our method.

as "OSSID (w/ Conf. Filter)" in Table I. In this setting, the input image I will be cropped to the region I' for pose estimation only if the detected bounding boxes have high predicted confidence, as defined by a detection score above a given threshold. Otherwise, if the detection score is below the defined threshold, pose estimation will be done using the full image I, resulting in a longer processing time. This confidence filtering is helpful in the early stage of online learning, where the detector may still have poor performance; if we don't use confidence filtering in such cases, the pose estimator will still run on the cropped image from the bounding box of a poor detector, which may cause the pose estimator to focus on the wrong region of the image. Confidence filtering will reject such low-confidence detections, and instead run the pose estimator on the entire observation. This leads to slower inference speed on average but better performance, as we can see in Table I.

In addition, we further study the potential negative effects of low-accuracy pose estimates in the pseudo ground truth used in the self-supervised training process, as these bad pose estimates may mislead the detector. We conduct an experiment where the ground truth object pose is used to finetune the detector. Specifically, we modify the online self-supervised protocol to use the ground truth bounding box and mask, instead of the estimated pose h, to finetune the detection network (line 7 of Algorithm 1). The results are reported in the "OSSID (Oracle)" column in Table I. The difference between "OSSID (Ours)" and "OSSID (Oracle)" shows there is still a gap between using the pseudo and real ground truth. However, labeling such high-quality ground truth requires extensive human efforts; in contrast, our self-supervised learning pipeline demonstrates a way of improving detection without manual labeling.

To evaluate the gap that still exists between the zero-shot and object-specific methods, we also report the performance of CosyPose [4] in Table I from the "CosyPose-ECCV20-PBR-1VIEW" variant on the BOP leaderboard[1]. This method is a state-of-the-art non-zero-shot pose estimator that is trained solely on synthetic data. However, CosyPose requires tens of GPU-days for synthetic data generation and network training. As shown in Table I, CosyPose (non-zero-shot, trained for tens of GPU-days on a much larger dataset), achieved better results than our approach in detection mAP and similar results to our approach in terms of pose AR score. However, such methods require either large-scale manual data annotation or synthetic data generation to work on new objects, with long data generation and training times, whereas our method can quickly adapt online to new objects. Further, our method can train directly on real data in a self-supervised way, without requiring manual annotations or synthetic dataset generation.

E. Transductive Learning

The performance can be further improved if it is possible to obtain all testing images beforehand and train the network offline with self-supervision. In such scenarios we devise a transductive learning pipeline [54], where the detector is self-supervised trained and then tested on the same set of testing images (without any annotations). In the offline

[1]https://bop.felk.cvut.cz/leaderboards/

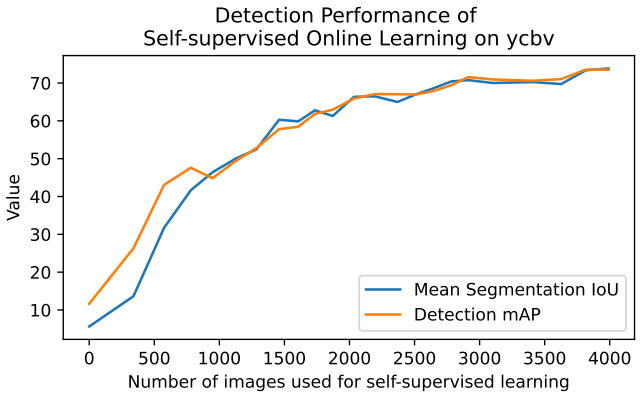


Fig. 5. Progress of self-supervised learning as more unlabeled test images are observed. The detection models are taken from different timestamps in online learning on the YCB-V dataset and the numbers are evaluated on the entire testing set. The detection and segmentation accuracies improve over time as a result of our online and self-supervised pipeline.

training stage, the zero-shot pose estimator runs on all testing images first and then the zero-shot detector is self-supervised trained on the pose estimation results (for 50 epochs in our experiments). This offline training process takes less than half an hour on the LM-O and YCB-V BOP testing set. The result of this transductive learning pipeline is shown as "OSSID (Transductive)" in Table I; we can see that transductive learning leads to another significant performance boost compared to the online learning pipeline on both datasets. Although this setup may not be used for real-time pose estimation, it can be used for estimating the poses of objects from a fixed dataset. Further, it provides an upper-bound performance of our method, allowing our method to learn from both past and future frames (instead of only learning from past frames).

F. Time Analysis

The significant speedup for zero-shot pose estimation results from shrinking the pose search space from the full observation image to just the region within the highest-scoring bounding box of the detector. We show the inference time breakdown of the pose estimator on the LM-O dataset in Fig. 4. We can see that, although a detection overhead of 50 milliseconds is added compared to ZePHyR, the time spent on pose hypothesis generation is reduced by a factor of 25. The online finetuning in total takes 6 minutes for the LM-O dataset and about one hour and a half for the YCB-V dataset on a single GPU, which is much less than tens of GPU-days needed to train object-specific detectors. The gradient updates can be run in a background thread and thus would not delay the inference time.

G. Online Learning

To further quantify how the model improves over the course of self-supervised training, we evaluate the model at different points of the OSSID pipeline. In Fig. 5, we take the model weights at different timestamps in self-supervised learning and evaluate them on the entire testing set. We report both mAP for the detection output and mean Intersection over Union (IoU) for the segmentation output from the detection network. It can be seen that as the network are self-supervised trained on more and more images of the scene, the accuracy of the detection increases. The results demonstrate that, through self-supervision, the instance detector gradually adapts to new objects and environments using the new observations, without the need for annotations. This opens up future directions of applying this method to robot manipulation tasks where the perception system needs to adapt to novel environments.

V. Conclusions

We propose a novel method, OSSID, using a slow zero-shot pose estimator to train a fast detection algorithm, without the need for any annotations. We show that a detector, trained in this self-supervised manner, shows adaptation to new objects and new environments, and exceeds the accuracy of similar zero-shot methods on cluttered environments. Further, we show that this detector can be used to filter the search space of a zero-shot pose estimator. This drastically reduces the inference time of the pose estimation system, while maintaining state-of-the-art accuracy. Our method thus shows the benefit of online self-supervised learning, resulting in a high-performance real-time pose estimation system that can be trained within 6 minutes (for the LM-O dataset).

VI. Acknowledgements

This work was supported in part by the National Science Foundation Smart and Autonomous Systems Program under Grant IIS-1849154 and in part by LG Electronics.

VII. Appendix

A. Implementation Details

In this section, we will provide more details about the implementation of the algorithms described in Sec. III.

1) Online Self-supervised Learning: The online self-supervised learning pipeline is described in Sec. III-C and Algorithm I. Here we will provide more details for the algorithm. The `Finetune()` function is called after every 32 finetuning examples of (I, h) are added to the finetuning set F. Within each `Finetune()`, the detection network is finetuned on the all examples in F for a single epoch. Similar to [33], we use AMS-Grad [55] for optimization with a learning rate of 10^{-4}, a weight decay of 10^{-6} and a batch size of 8.

2) Transductive Learning: The transductive learning pipeline simulates the scenarios where all the testing images are known before the network makes any inference and thus the network can be self-supervised trained on the testing images with more time budgets. Specifically, we run the zero-shot pose estimator on the uncropped images in the testing dataset first and use the pose estimates to train the detector in a regular epoch training fashion. In this way the pose estimation stage is the same as ZePHyR [9] and we can simply take the results from [9] as pseudo ground truth in the implementation. For training, we initialize the weights

of the detector as described in Sec. IV-A and finetune it for 50 epochs on the pseudo ground truth. We use the same optimizer as in the online learning pipeline, and shrink the learning rate to its tenth at the epoch 20 and 40.

Here we further report the training time using the transductive learning protocol. Since the network is only trained on the testing dataset, which contains 1445 images for LM-O and 4123 images for YCB-V. Therefore the network training for transductive learning only takes 50 epochs is roughly 25 minutes for LM-O and 72 minutes for YCB-V. This demonstrates that the proposed pipeline is capable to adapt to new environments and novel objects in a short time.

B. Synthetic Data Generation and Training

Since YCB-V objects were already used for training in [33], we need to re-train the DTOID network using another dataset in order to show the generalization ability to novel objects. Therefore we generated a synthetic dataset using BlenderProc [56]. We adopted the objects from BOP datasets [5] except for those from LM, LM-O and YCB-V. and additionally used 200 ShapeNet objects [57] with randomized CC0 textures [58]. The scenes were generated by randomly dropping objects onto a table and images were captured at randomly sampled camera poses. In this way, we produced a dataset of 40,000 images and trained the detection network from random weights for 100 epochs. We used the same optimizer and scheduler as described in Sec. VII-A.2.

Note that our DTOID weights did not reproduce the zero-shot detection performance as reported in [33]. However, the performance of our DTOID model quickly adapts to YCB-V objects as shown in Sec. IV-D and Table I.

C. Comparison to Non Zero-shot Detector

To analyze the benefit of zero-shot networks in our pipeline, we tested our self-supervised learning framework where the DTOID detection network is replaced by a a non zero-shot detector, specifically Mask R-CNN [59] with the ResNet-50 [60] backbone pretrained on MS COCO dataset [61]. The results on the LM-O dataset are shown in Table III. We found that in the transductive learning setting, Mask R-CNN can yield similar pose estimation performance as DTOID, but worse detection results. In the online learning setting, DTOID shows much better performance than Mask R-CNN. The reason might be that we need a much larger dataset to train a non zero-shot object detector, while zero-shot detectors like DTOID are designed to quickly adapt to new objects.

D. Qualitative Results

In Fig. 6, we show some qualitative results of the detector during the progress of the online self-supervised learning pipeline. Here we recorded the model weights after it is trained on different portion of the test dataset and compare their performance. We can see that the performance gradually improves and the previously missed or false detection are corrected as the online self-supervised learning continues.

Method	Task	Ours	Ours w/ Mask R-CNN
Online Learning	Detection mAP	64.0	36.5
	Pose AR score	61.7	43.9
Transductive Learning	Detection mAP	78.4	56.1
	Pose AR score	66.0	62.7

TABLE III

COMPARISON OF OUR RESULTS AND AN ABLATION REPLACING DTOID WITH MASK R-CNN. RESULTS ARE REPORTED ON THE LM-O DATASET.

REFERENCES

[1] T. Hodaň, V. Vineet, R. Gal, E. Shalev, J. Hanzelka, T. Connell, P. Urbina, S. Sinha, and B. Guenter, "Photorealistic image synthesis for object instance detection," in *ICIP*, 2019, pp. 66–70.

[2] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *RSS*, 2018.

[3] Z. Li, G. Wang, and X. Ji, "CDPN: Coordinates-Based disentangled pose network for Real-Time RGB-Based 6-DoF object pose estimation," in *ICCV*, 2019, pp. 7677–7686.

[4] Y. Labbe, J. Carpentier, M. Aubry, and J. Sivic, "Cosypose: Consistent multi-view multi-object 6d pose estimation," in *ECCV*, 2020, pp. 574–591.

[5] T. Hodaň, F. Michel, E. Brachmann, W. Kehl, A. Glent Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, J. Matas, and C. Rother, "BOP: Benchmark for 6D object pose estimation," *ECCV*, pp. 19–35, 2018.

[6] Y. Xiao, X. Qiu, P. Langlois, M. Aubry, and R. Marlet, "Pose from shape: Deep pose estimation for arbitrary 3d objects," in *BMVC*, 2019, p. 61.

[7] K. Park, A. Mousavian, Y. Xiang, and D. Fox, "Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation," in *CVPR*, 2020, pp. 10 707–10 716.

[8] M. Sundermeyer, M. Durner, E. Y. Puang, Z.-C. Marton, N. Vaskevicius, K. O. Arras, and R. Triebel, "Multi-path learning for object pose estimation across domains," in *CVPR*, 2020, pp. 13 913–13 922.

[9] B. Okorn, Q. Gu, M. Hebert, and D. Held, "Zephyr: Zero-shot pose hypothesis rating," in *ICRA*, 2021, pp. 14 141–14 148.

[10] T. Weise, B. Leibe, and L. Van Gool, "Accurate and robust registration for in-hand modeling," in *CVPR*, 2008, pp. 1–8.

[11] Q.-Y. Zhou and V. Koltun, "Dense scene reconstruction with points of interest," *ACM ToG*, vol. 32, no. 4, pp. 1–8, 2013.

[12] T. Weise, T. Wismer, B. Leibe, and L. Van Gool, "Online loop closure for real-time interactive 3D scanning," *CVIU*, pp. 635–648, 2011.

[13] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in hand model acquisition," in *ICRA*, 2010, pp. 1817–1824.

[14] F. Wang and K. Hauser, "In-hand object scanning via RGB-D video segmentation," in *ICRA*, 2019, pp. 3296–3302.

[15] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *CVPR*, 2010, pp. 998–1005.

[16] B. Drost and S. Ilic, "3d object detection and localization using multimodal point pair features," in *3DIMPVT*, 2012, pp. 9–16.

[17] E. Kim and G. Medioni, "3d object recognition in range images using visibility context," in *IROS*, 2011, pp. 3800–3807.

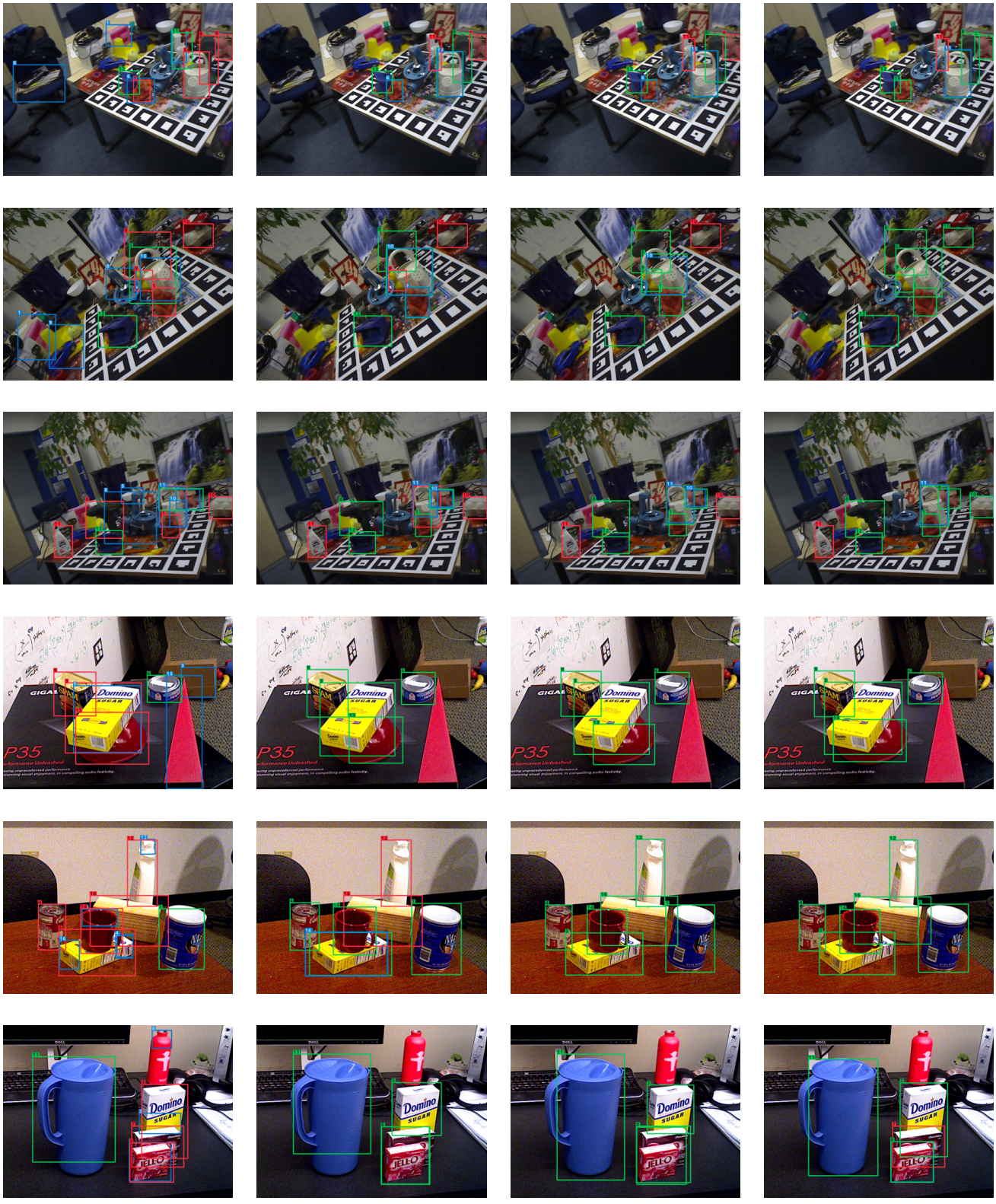
[18] S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige, "Going further with point pair features," in *ECCV*, 2016, pp. 834–848.

[19] J. Vidal, C.-Y. Lin, X. Lladó, and R. Martí, "A method for 6d pose estimation of free-form rigid objects using point pair features on range data," *Sensors*, vol. 18, no. 8, p. 2678, 2018.

[20] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6d object pose estimation using 3d object coordinates," in *ECCV*, 2014, pp. 536–551.

[21] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *CVPR*, 2019, pp. 3343–3352.

[22] R. König and B. Drost, "A hybrid approach for 6dof pose estimation," in *ECCV*, 2020, pp. 700–706.



(a) Initial model (b) After 1/3 images (c) After 2/3 images (d) After all images

Fig. 6. Qualitative results of the online self-supervised learned detector, after seeing different number of images. Green, blue and red boxes mean correct, false and missed detection results respectively. The leftmost column shows the results of the baseline detection model and the others show the performance after the detector has been trained on different portion of the test set. The first three rows are the results from the LM-O dataset and the last three rows are from the YCB-V dataset.

[23] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-Shot object detection via feature reweighting," in *ICCV*, 2019, pp. 8419–8428.

[24] X. Wang, T. E. Huang, J. Gonzalez, T. Darrell, and F. Yu, "Frustratingly simple few-shot object detection," in *ICML*, 2020, pp. 9919–9928.

[25] H. Wang, X. Zhang, Y. Hu, Y. Yang, X. Cao, and X. Zhen, "Few-Shot semantic segmentation with democratic attention networks," in *ECCV*, 2020, pp. 730–746.

[26] M. Boudiaf, H. Kervadec, I. M. Ziko, P. Piantanida, I. B. Ayed, and J. Dolz, "Few-shot segmentation without meta-learning: A good transductive inference is all you need?" in *CVPR*, 2021, pp. 13 979–13 988.

[27] Q. Fan, W. Zhuo, C.-K. Tang, and Y.-W. Tai, "Few-shot object detection with attention-RPN and multi-relation detector," in *CVPR*, 2020, pp. 4013–4022.

[28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[29] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *CVPR*, 2003, pp. 264–271.

[30] A. Collet, M. Martinez, and S. S. Srinivasa, "The moped framework: Object recognition and pose estimation for manipulation," *IJRR*, vol. 30, no. 10, pp. 1284–1306, 2011.

[31] J. Tang, S. Miller, A. Singh, and P. Abbeel, "A textured object recognition pipeline for color and depth image data," in *ICRA*, 2012, pp. 3467–3474.

[32] P. Ammirato, C.-Y. Fu, M. Shvets, J. Kosecka, and A. C. Berg, "Target driven instance detection," *arXiv:1803.04610 [cs]*, Oct. 2019.

[33] J.-P. Mercier, M. Garon, P. Giguère, and J.-F. Lalonde, "Deep template-based object instance detection," in *WACV*, 2021, pp. 1506–1515.

[34] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *CVPR*, 2018, pp. 3339–3348.

[35] Z. He and L. Zhang, "Multi-adversarial faster-rcnn for unrestricted object detection," in *ICCV*, 2019, pp. 6668–6677.

[36] H.-K. Hsu, C.-H. Yao, Y.-H. Tsai, W.-C. Hung, H.-Y. Tseng, M. Singh, and M.-H. Yang, "Progressive domain adaptation for object detection," in *WACV*, 2020, pp. 749–757.

[37] M. Xu, H. Wang, B. Ni, Q. Tian, and W. Zhang, "Cross-domain detection via graph-induced prototype alignment," in *CVPR*, 2020, pp. 12 355–12 364.

[38] V. Vs, V. Gupta, P. Oza, V. A. Sindagi, and V. M. Patel, "MeGA-CDA: Memory guided attention for Category-Aware unsupervised domain adaptive object detection," in *CVPR*, 2021, pp. 4516–4526.

[39] P. Oza, V. A. Sindagi, V. S. Vibashan, and V. M. Patel, "Unsupervised domain adaptation of object detectors: A survey," *arXiv:2105.13502 [cs]*, May 2021.

[40] A. RoyChowdhury, P. Chakrabarty, A. Singh, S. Jin, H. Jiang, L. Cao, and E. Learned-Miller, "Automatic adaptation of object detectors to new domains using self-training," in *CVPR*, 2019, pp. 780–790.

[41] M. Khodabandeh, A. Vahdat, M. Ranjbar, and W. G. Macready, "A robust learning approach to domain adaptive object detection," in *ICCV*, 2019, pp. 480–490.

[42] S. Kim, J. Choi, T. Kim, and C. Kim, "Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection," in *ICCV*, 2019, pp. 6091–6100.

[43] S. Pirk, M. Khansari, Y. Bai, C. Lynch, and P. Sermanet, "Online object representations with contrastive learning," *arXiv:1906.04312 [cs]*, June 2019.

[44] C. Mitash, K. E. Bekris, and A. Boularias, "A self-supervised learning system for object detection using physics simulation and multi-view pose estimation," in *IROS*, 2017, pp. 545–551.

[45] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999, pp. 1150–1157.

[46] itSeez3D, "#1 mobile 3d scanning app for ipad — itseez3d," https://itseez3d.com/scanner.html, accessed: 2021-11-19.

[47] E. V. T. LTD, "Qlone, 3d scan any object, anywhere!" https://www.qlone.pro/, accessed: 2021-11-19.

[48] Trnio, "Trnio 3d scanner," https://www.trnio.com/, accessed: 2021-11-19.

[49] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *ACCV*, 2013, pp. 548–562.

[50] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *ICAR*, 2015, pp. 510–517.

[51] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.

[52] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *ICCV*, 2011, pp. 858–865.

[53] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *CVPR*, 2019, pp. 1328–1338.

[54] A. Arnold, R. Nallapati, and W. W. Cohen, "A comparative study of methods for transductive transfer learning," in *ICDM Workshops*, 2007, pp. 77–82.

[55] J. R. Sashank, K. Satyen, and K. Sanjiv, "On the convergence of adam and beyond," in *ICLR*, vol. 5, 2018, p. 7.

[56] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam, "BlenderProc," *arXiv:1911.01911 [cs]*, Oct. 2019.

[57] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D model repository," *arXiv:1512.03012 [cs]*, Dec. 2015.

[58] ambientCG, "ambientCG - free public domain PBR materials," https://ambientcg.com/, accessed: 2021-9-20.

[59] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2961–2969.

[60] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[61] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *ECCV*, 2014, pp. 740–755.