HAL
open science

MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare

Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, Josef Sivic

To cite this version:

Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, et al.. MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare. CoRL 2022 - Conference on Robot Learning, Dec 2022, Auckland, New Zealand. hal-03910329

HAL Id: hal-03910329

https://hal.science/hal-03910329v1

Submitted on 22 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare

Yann Labbé [1†] **Lucas Manuelli** [2] **Arsalan Mousavian** [2] **Stephen Tyree** [2]
Stan Birchfield [2] **Jonathan Tremblay** [2] **Justin Carpentier** [1] **Mathieu Aubry** [3]
Dieter Fox [2,4] **Josef Sivic** [5]
[1] ENS/Inria [2] NVIDIA [3] LIGM/ENPC [4] University of Washington [5] CIIRC CTU
megapose6d.github.io

Abstract: We introduce MegaPose, a method to estimate the 6D pose of novel objects, that is, objects unseen during training. At inference time, the method only assumes knowledge of (i) a region of interest displaying the object in the image and (ii) a CAD model of the observed object. The contributions of this work are threefold. First, we present a 6D pose refiner based on a render & compare strategy which can be applied to novel objects. The shape and coordinate system of the novel object are provided as inputs to the network by rendering multiple synthetic views of the object's CAD model. Second, we introduce a novel approach for coarse pose estimation which leverages a network trained to classify whether the pose error between a synthetic rendering and an observed image of the same object can be corrected by the refiner. Third, we introduce a large scale synthetic dataset of photorealistic images of thousands of objects with diverse visual and shape properties, and show that this diversity is crucial to obtain good generalization performance on novel objects. We train our approach on this large synthetic dataset and apply it *without retraining* to hundreds of novel objects in real images from several pose estimation benchmarks. Our approach achieves state-of-the-art performance on the ModelNet and YCB-Video datasets. An extensive evaluation on the 7 core datasets of the BOP challenge demonstrates that our approach achieves performance competitive with existing approaches that require access to the target objects during training. Code, dataset and trained models are available on the project page [1].

1 Introduction

Accurate 6D object pose estimation is essential for many robotic and augmented reality applications. Current state-of-the-art methods are learning-based [2, 3, 4, 5, 6] and require 3D models of the objects of interest at both training and test time. These methods require hours (or days) to generate synthetic data for each object and train the pose estimation model. They thus cannot be used in the context of robotic applications where the objects are only known during inference (e.g. CAD models are provided by a manufacturer or reconstructed [7]), and where rapid deployment to novel scenes and objects is key.

The goal of this work is to estimate the 6D pose of novel objects, *i.e.*, objects that are only available at *inference time* and are not known in advance during training. This problem presents the challenge of generalizing to the large variability in shape, texture, lighting conditions, and severe occlusions that can be encountered in real-world applications. Some prior works [8, 9, 10, 11, 12, 13, 14] have considered category-level pose estimation to partially address the challenge of novel objects by developing methods that can generalize to novel object instances of a known class (e.g. mugs or shoes). These methods however do not generalize to object instances outside of training categories.

[1] Inria Paris and Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, 75005 Paris, France.
[3] LIGM, École des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-vallée, France.
[5] Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.
[†] Work partially done while the author was an intern at NVIDIA.

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

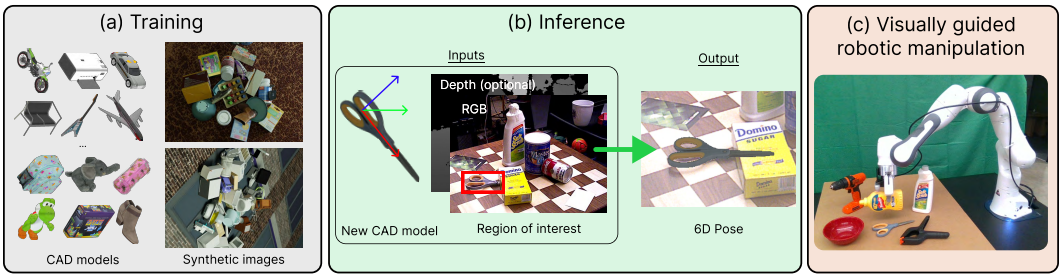

Figure 1: MegaPose is a 6D pose estimation approach (a) that is trained on millions of synthetic scenes with thousands of different objects and (b) can be applied *without re-training* to estimate the pose of any novel object, given a CAD model and a region of interest displaying the object. It can thus be used to rapidly deploy visually guided robotic manipulation systems in novel scenes containing novel objects (c).

Other methods aim at generalizing to any novel instances regardless of their category [15, 16, 17, 18, 19, 20, 21, 22]. These works present important technical limitations. They rely on non-learning based components for generating pose hypotheses [21] (e.g. PPF [23]), for pose refinement [17] (e.g. PnP [24] and ICP [25, 26]), for computing photometric errors in pixel space [15], or for estimating the object depth [18, 16] (e.g. using only the size of a 2D detection [27]). These components however inherently cannot benefit from being trained on large amount of data to gain robustness with respect to noise, occlusions, or object variability. Learning-based methods also have the potential to improve as the quality and size of the datasets improve.

Pipelines for 6D pose estimation of known (not novel) objects that consist of multiple learned stages [4, 5] have shown excellent performance on several benchmarks [2] with various illumination conditions, textureless objects, cluttered scenes and high levels of occlusions. We take inspiration from [4, 5] which split the problem into three parts: (i) 2D object detection, (ii) coarse pose estimation, and (iii) iterative refinement via render & compare. We aim at extending this approach to novel objects unseen at training time. The detection of novel objects has been addressed by prior works [17, 28, 29, 30] and is outside the scope of this paper. In this work, we focus on the coarse and refinement networks for 6D pose estimation. Extending the paradigm from [4] presents three major challenges. First, the pose of an object depends heavily on both its visual appearance and choice of coordinate system (defined in the CAD model of the object). In existing refinement networks based on render & compare [20, 4], this information is encoded in the network weights during training, leading to poor generalization results when tested on novel objects. Second, direct regression methods for coarse pose estimation are trained with specific losses for symmetric objects [4], requiring that object symmetries be known in advance. Finally, the diversity of shape and visual properties of the objects that can be encountered in real-world applications is immense. Generalizing to novel objects requires robustness to properties such as object symmetries, variability of object shape, and object textures (or absence of).

Contributions. We address these challenges and propose a method for estimating the pose of any novel object in a single RGB or RGB-D image, as illustrated in Figure 1. First, we propose a novel approach for 6D pose refinement based on render & compare which enables generalization to novel objects. The shape and coordinate system of the novel object are provided as inputs to the network by rendering multiple synthetic views of the object's CAD model. Second, we propose a novel method for coarse pose estimation which does not require knowledge of the object symmetries during training. The coarse pose estimation is formulated as a classification problem where we compare renderings of random pose hypotheses with an observed image, and predict whether the pose can be corrected by the refiner. Finally, we leverage the availability of large-scale 3D model datasets to generate a highly diverse synthetic dataset consisting of 2 million photorealistic [31] images depicting over 20K models in physically plausible configurations. The code, dataset and trained models are available on the project page [1].

We show that our novel-object pose estimation method trained on our large-scale synthetic dataset achieves state-of-the-art performance on ModelNet [32, 20]. We also perform an extensive evaluation of the approach on hundreds of novel objects from all 7 core datasets of the BOP challenge [2]

and demonstrate that our approach achieves performance competitive with existing approaches that require access to the target objects during training.

2 Related work

In this section, we first review the literature on 6D pose estimation of known rigid objects. We then focus on the practical scenario similar to ours where the objects are not known prior to training.

6D pose estimation of known objects. Estimating the 6D pose of rigid objects is a fundamental computer vision problem [33, 34, 35] that was first addressed using correspondences established with locally invariant features [35, 36, 37, 38, 23] or template matching [39, 40]. These have been replaced by learning-based methods with convolutional neural networks that directly regress sets of sparse [41, 42, 27, 43, 44, 45, 46] or dense [47, 48, 49, 50, 3, 44] features. All these approaches use non-learning stages relying on PnP+Ransac [51, 24] to recover the pose from correspondences in RGB images, or variants of the iterative closest point algorithm, ICP [25, 26], when depth is available. The best performing methods rely on trainable refinement networks [52, 20, 4, 20, 5] based on render & compare [53, 54, 55, 20]. These methods render a single image of the object, which is not sufficient to provide complete information on the shape and coordinate system of a 3D model to the network. This information is thus encoded in the networks weights when training, which leads to poor generalization when tested on novel objects unseen at training. Our approach renders multiple views of an object to provide this 3D information, making the trained network independent of these object-specific properties.

6D pose estimation of novel objects. Other works consider a practical scenario where the objects are not known in advance. Category-level 6D pose estimation is a popular problem [8, 9, 10, 11, 12, 13, 14] in which CAD models of test objects are not known, but the objects are assumed to belong to a known category. These methods rely on object properties that are common within categories (*e.g.* handle of a mug) to define and estimate the object pose, and thus cannot generalize to novel categories. Our method requires the 3D model of the novel object instance to be known during inference, but does not rely on any category-level information. Other works address a scenario similar to ours. [56, 19, 57, 18, 16, 30] only estimate the 3D orientation of novel objects by comparing rendered pose hypotheses with the observed image using features extracted by a network. They rely on handcrafted [18, 16] or learning-based DeepIM [19] refiners to recover accurate 6D poses. We instead propose a method that estimates the full 6D pose of the object and show our refinement network significantly outperforms DeepIM [20] when tested on novel object instances. The closest works to ours are OSOP [17] and ZePHyR [21]. OSOP focuses on the coarse estimation by explicitly predicting 2D-2D correspondences between a single rendered view of the object and the observed image, and solves for the pose using PnP or Kabsch [25] which makes inference slower and less robust compared to directly predicting refinement transforms with a network as done in our solution. ZePHyR [21] strongly relies on the depth modality, whereas our approach can also be used in RGB-only images. Finally, [15, 58, 22, 59] investigate using a set of real reference views of the novel object instead of using a CAD model. These approaches have only reported results on datasets with limited or no occlusions. Our use of a deep render & compare network trained on a large-scale synthetic dataset displaying highly occluded object instances enables us to handle highly cluttered scenes with high occlusions like in the LineMOD Occlusion, HomebrewedDB or T-LESS datasets.

3 Method

In this section we present our framework for pose estimation of novel objects. Our goal is to detect the pose $\mathcal{T}_{CO}$ (the pose of object frame O expressed in camera frame C composed of 3D rotation and 3D translation) of a novel object given an input RGB (or RGBD) image, I_o, and a 3D model of the object. Similar to DeepIM [20] and CosyPose [4], our method consists of three components (1) object detection, (2) coarse pose estimation and (3) pose refinement. Compared to these works, our proposed method enables generalization to novel objects not seen during training, requiring novel

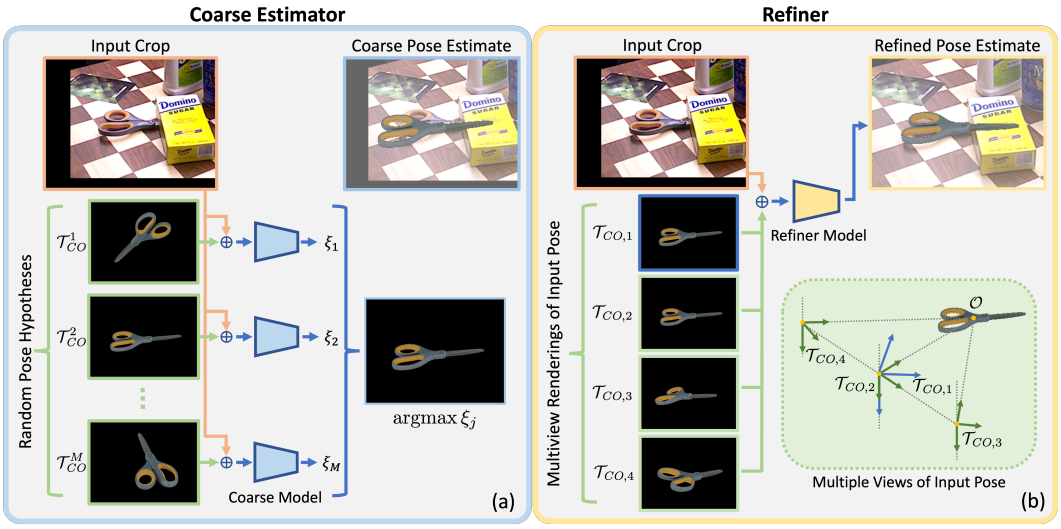

Figure 2: $\oplus$ denotes concatenation. **(a) Coarse Estimator:** Given a cropped input image the coarse module renders the object in multiple input poses $\{\mathcal{T}_{CO}^j\}$. The coarse network then classifies which rendered image best matches the observed image. **(b) Refiner:** Given an initial pose estimate $\mathcal{T}_{CO}^k$ the refiner renders the objects at the estimated pose $\mathcal{T}_{CO,1} := \mathcal{T}_{CO}^k$ (blue axes) along with 3 additional viewpoints $\{\mathcal{T}_{CO,i}\}_{i=2}^4$ (green axes) defined such that the camera z-axis intersects the anchor point $\mathcal{O}$. The refiner network consumes the concatenation of the observed and rendered images and predicts an updated pose estimate $\mathcal{T}_{CO}^{k+1}$.

approaches for the coarse model, the refiner and the training data. Our approach can accept either RGB or RGBD inputs, if depth is available the RGB and D images are concatenated before being passed into the network. Detection of novel-objects in an image is an interesting problem that has been addressed in prior work [28, 60, 17, 22, 30] but lies outside the scope of this paper. Thus for our experiments we assume access to an object detector, but emphasize that our method can be coupled with any object detector, including zero-shot methods such as those in [28, 60].

3.1 Technical Approach

Coarse pose estimation. Given an object detection, shown in Figure 1(b), the goal of the coarse pose estimator is to provide an initial pose estimate $\mathcal{T}_{CO,\text{coarse}}$ which is sufficiently accurate that it can then be further improved by the refiner. In order to generalize to novel-objects we propose a novel classification based approach that compares observed and rendered images of the object in a variety of poses and selects the rendered image whose object pose best matches the observed object pose.

Figure 2(a) gives an overview of the coarse model. At inference time the network consumes the observed image I_o along with rendered images $\{I_r(\mathcal{T}_{CO}^j)\}_{j=1}^M$ of the object in many different poses $\{\mathcal{T}_{CO}^j\}_{j=1}^M$. For each pose $\mathcal{T}_{CO}^j$ the model predicts a score $(I_o, I_r(\mathcal{T}_{CO}^j)) \rightarrow \xi_j$ that classifies whether the pose hypothesis is within the basin of attraction of the refiner. The highest scoring pose $\mathcal{T}_{CO}^{j^*}, j^* = \text{argmax}_j\, \xi_j$ is used as the initial pose for the refinement step. Since we are performing classification, our method can implicitly handle object symmetries, as multiple poses can be classified as correct.

Pose refinement model. Given an input image and an estimated pose, the refiner predicts an updated pose estimate. Starting from a coarse initial pose estimate $\mathcal{T}_{CO,\text{coarse}}$ we can iteratively apply the refiner to produce an improved pose estimate. Similar to [4, 20] our refiner takes as input observed I_o and rendered images $I_r(\mathcal{T}_{CO}^k)$ and predicts an updated pose estimate $\mathcal{T}_{CO}^{k+1}$, see Figure 2 (b), where k refers to the kth iteration of the refiner. Our pose update uses the same parameterization as DeepIM [20] and CosyPose [4] which disentangles rotation and translation prediction. Crucially this pose update $\Delta\mathcal{T}$ depends on the choice of an *anchor point* $\mathcal{O}$, see Appendix for more details. In prior work [4, 20] which trains and tests on the same set of objects, the network can effectively learn

the position of the anchor point $\mathcal{O}$ for each object. However in order to generalize to novel objects we must enable the network to infer the anchor point $\mathcal{O}$ at inference time.

In order to provide information about the anchor point to the network we always render images $I_r(\mathcal{T}_{CO}^k)$ such that the anchor point $\mathcal{O}$ projects to the image center. Using rendered images from multiple distinct viewpoints $\{\mathcal{T}_{CO,i}\}_{i=1}^N$ the network can infer the location of the anchor point $\mathcal{O}$ as the intersection point of camera rays that pass through the image center, see Figure 2(b).

Additional information about object shape and geometry can be provided to the network by rendering depth and surface normal channels in the rendered image I_r. We normalize both input depth (if available) and rendered depth images using the currently estimated pose $\mathcal{T}_{CO}^k$ to assist the network in generalizing across object scales, see Appendix for more details.

Network architecture. Both the coarse and refiner networks consists of a ResNet-34 backbone followed by spatial average pooling. The coarse model has a single fully-connected layer that consumes the backbone feature and outputs a classification logit. The refiner network has a single fully-connected layer that consumes the backbone feature and outputs 9 values that specify the translation and rotation for the pose update.

3.2 Training Procedure

Training data. For training, both the coarse and refiner models require RGB(-D)[1] images with ground-truth 6D object pose annotations, along with 3D models for these objects. In order for our approach to generalize to novel-objects we require a large dataset containing diverse objects. All of of our methods are trained purely on synthetic data generated using BlenderProc [31]. We generate a dataset of 2 million images using a combination of ShapeNet [61] (abbreviated as SN) and Google-Scanned-Objects (abbreviated as GSO) [7]. Similar to the BOP [62] synthetic data, we randomly sampled objects from our dataset and dropped them on a plane using a physics simulator. Materials, background textures, lighting and camera positions are randomized. Example images can be seen in Figure 1(a) and in the Appendix. Some of our ablations also use the synthetic training datasets provided by the BOP challenge [62]. We add data augmentation similar to CosyPose [4] to the RGB images which was shown to be a key to successful sim-to-real transfer. We also apply data augmentation to the depth images as explained in the appendix.

Refiner model. The refiner model is trained similarly to [4]. Given an image with an object $\mathcal{M}$ at ground-truth pose $\mathcal{T}_{CO}^*$ we generate a perturbed pose $\mathcal{T}_{CO}'$ by applying a random translation and rotation to $\mathcal{T}_{CO}^*$. Translation is sampled from a normal distribution with a standard deviations of $(0.02, 0.02, 0.05)$ centimeters and rotation is sampled as random Euler angles with a standard deviation of 15 degrees in each axis. The network is trained to predict the relative transformation between the initial and target pose. Following [4, 20] we use a loss that disentangles the prediction of depth, x-y translation, and rotation. See the appendix for more details.

Coarse model. Given an input image I_o of an object $\mathcal{M}$ and a pose $\mathcal{T}_{CO}'$ the coarse model is trained to classify whether pose $\mathcal{T}_{CO}'$ is within the basin of attraction of the refiner. In other words, if the refiner were started with the initial pose estimate $\mathcal{T}_{CO}'$ would it be able to estimate the ground-truth pose via iterative refinement? Given a ground-truth pose-annotation $\mathcal{T}_{CO}^*$ we randomly sample poses $\mathcal{T}_{CO}'$ by adding random translation and rotation to $\mathcal{T}_{CO}^*$. The positives are sampled from the same distribution used to generate the perturbed poses the refiner network is trained to correct (see above), and other poses sufficiently distinct to this one (see the appendix for more details) are marked as negatives. The model is then trained with binary cross entropy loss.

4 Experiments

We evaluate our method for 6D pose estimation of novel objects using the seven challenging datasets of the BOP [2, 62] 6D pose estimation benchmark, and the ModelNet [20] dataset. The dataset and

[1] Our method can consume either RGB or RGB-D images depending on the input modalities that are available.

	Pose Initialization		Pose Refinement			BOP Datasets							
	Method	Novel objects	Method	Novel objects	RGB-D Input	LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	Mean
1	CosyPose [4]	✗	CosyPose	✗		63.3	64.0	68.5	58.3	21.6	65.6	57.4	57.0
2	SurfEmb [3]	✗	BFGS	✗		66.3	73.5	71.5	58.8	41.3	79.1	64.7	65.0
3	SurfEmb [3]	✗	BFGS+ICP	✓	✓	*75.8*	*82.8*	*85.4*	*65.6*	*49.8*	*86.7*	*80.6*	*75.2*
4	OSOP [17]	✓	Multi-Hyp.	✓		31.2	-	-	-	-	49.2	33.2	-
5	OSOP [17]	✓	MH+ICP	✓	✓	48.2	-	-	-	-	60.5	57.2	-
6	(PPF, Sift) + Zephyr [21]	✓	-	✓	✓	**59.8**	-	-	-	-	-	51.6	-
7	(PPF, Sift) + Our coarse	✓	Our refiner	✓	✓	57.0	-	-	-	-	-	62.3	-
8	CosyPose [4]	✗	–			53.6	52.0	57.6	53.0	13.1	33.5	33.3	42.3
9	CosyPose [4]	✗	Ours	✓		65.5	72.0	70.1	57.3	28.4	67.0	56.8	59.6
10	CosyPose [4]	✗	Ours	✓	✓	71.2	63.8	85.0	55.1	39.9	73.2	69.2	66.0
11	Ours	✓	–			18.7	19.7	20.5	15.3	8.00	18.6	13.9	16.2
12	Ours	✓	Ours	✓		53.7	**62.2**	58.4	**43.6**	30.1	72.9	60.4	54.5
13	Ours	✓	Ours	✓	✓	58.3	54.3	**71.2**	37.1	**40.4**	**75.7**	**63.3**	**57.2**

Table 1: Results on the BOP challenge datasets. We report the AR score on each of the 7 datasets considered in the BOP challenge and the mean score across datasets. With the exception of Zephyr (row 11), all approaches are trained purely on synthetic data. For each column, we denote the best over result in *italics* and the best novel-object pose estimation method in **bold**.

the standard 6D pose estimation metrics we use are detailed in Section 4.1. In all our experiments, the objects are considered novel, i.e. they are only available during inference on a new image and they are not used during training. In Section 4.2, we evaluate the performance of our approach composed of coarse and refinement networks. Notably, we show that (i) our method is competitive with others that require the object models to be known in advance, and (iii) our refiner outperforms current state-of-the-art on the ModelNet and YCB-V datasets. Section 4.3 validates our technical contributions and shows the crucial importance of the training data in the success of our method. Finally, we discuss the limitations in Section 4.4.

4.1 Dataset and metrics

We consider the seven core datasets of the BOP challenge [62, 2]: LineMod Occlusion (LM-O) [63], T-LESS [64], TUD-L [62], IC-BIN [65], ITODD [66], HomebrewedDB (HB) [67] and YCB-Video (YCB-V) [47]. These datasets exhibit 132 different objects in cluttered scenes with occlusions. These objects present many factors of variation: textured or untextured, symmetric or asymmetric, household or industrial (e.g. watcher pitcher, stapler, bowls, multi-socket plug adaptor) which makes them representative of objects that are typically encountered in robotic scenarios. The ModelNet dataset depicts individual instances of objects from 7 classes of the ModelNet [32] dataset (bathtub, bookshelf, guitar, range hood, sofa, tv stand and wardrobe). We use initial poses provided by adding noise to the ground truth, similar to previous works [20, 16, 15]. The focus is on refining these inital poses. We follow the evaluation protocol of [2] for BOP datasets, and of DeepIM [20] for ModelNet.

4.2 6D pose estimation of novel objects

Performance of coarse+refiner. Table 1 reports results of our novel-object pose estimation method on the BOP datasets. We first use the detections and pose hypotheses provided by a combination of PPF and SIFT, similar to the state-of-the-art method Zephyr [21]. For each object detection, these algorithms provide multiple pose hypotheses. We find the best hypothesis using the score of our coarse network, and apply 5 iterations of our refiner. Results are reported in row 7. On YCB-V, our method achieves a +10.7 AR score improvement compared to Zephyr (row 6). Averaged across the YCB-V and LM-O datasets, the AR score of our approach is 59.7 compared to 55.7 for Zephyr (row 6). Next, we provide a complete set of results using the detections from Mask-RCNN networks. Please note that since detection of novel objects is outside the scope of this paper, we use the networks trained on the synthetic PBR data of the target objects [2] which are publicly available for each dataset. We report the results of our coarse estimation strategy (Table 1, row 11), and after running the refiner network, on RGB (row 12) or RGB-D (row 13) images. We observe that (i) our refinement network

Method	RGB-D	Average Recall		
		(5°, 5cm)	ADD (0.1d)	Proj2D (5px)
DeepIM [20]	✓	64.3	83.6	73.3
Multi-Path [16]		84.8	90.1	81.6
LatentFusion[15]	✓	85.5	94.3	94.7
Ours		88.6	90.5	88.9
Ours	✓	**97.6**	**98.9**	**97.5**

Table 2: Evaluation of the refiner on the ModelNet [20] dataset. The mean average recall is computed over the seven classes of the dataset.

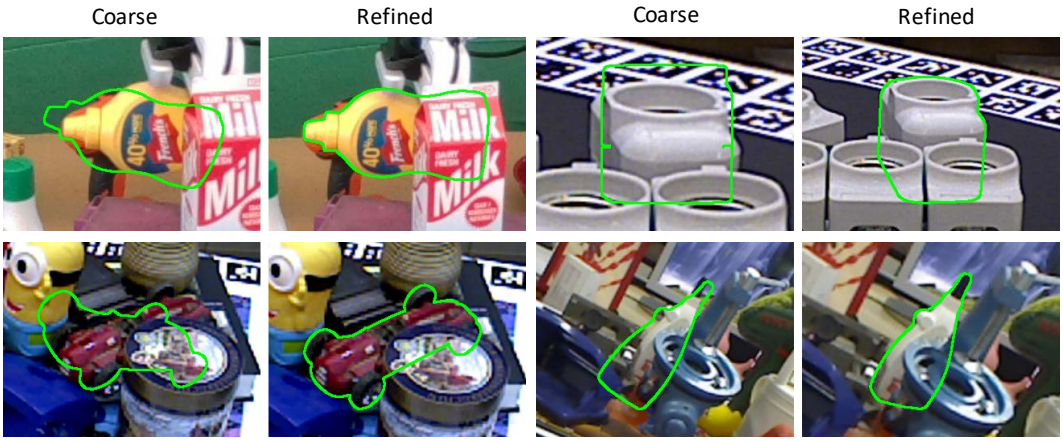

Figure 3: Qualitative results. For each pair of images, the left image is a visualization of our coarse estimate, and the right image is after applying 5 iterations of our refiner. None of these objects from the YCBV, LMO, HB, or T-LESS datasets were used for training our approach. Please notice the high accuracy of MegaPose despite (i) severe occlusions and (ii) the varying properties of the novel objects (e.g. the texture-less industrial plug in the top-right example, textured mustard bottle in the top-left).

significantly improves the coarse estimates (+41.0 mean AR score for our RGBD refiner) and (ii) the performance of both models is competitive with the learning-based refiner of CosyPose [4] (row 1) while not requiring to be trained on the test objects. The recent SurfEmb [3] performs better than our approach, but heavily relies on the knowledge of the objects for training and cannot generalize to novel objects.

Performance of the refiner. We now focus on the evaluation of our refiner which can be used to refine arbitrary initial poses. Our refiner is the only learning-based approach in Table 1 which can be applied to novel objects. In rows 9 and 10, we apply our refiner to the coarse estimates of CosyPose [4] (row 11). Again, we observe that our refiner significantly improves the accuracy of these initial pose estimates (+23.7 in average for the RGB-D model). Notably, the RGB-only method (row 9) performs better than the CosyPose refiner (row 1) on average, while not having seen the BOP objects during training. This is thanks to our large-scale training on thousands of various objects, while CosyPose is only trained on tens of objects for each dataset.

One iteration of our refiner takes approximately 50 milliseconds on a RTX 2080 GPU, making it suitable for use in an online tracking application. Five iterations of our refiner are also 5 times faster than the object-specific refiner of SurfEmb [3] which takes around 1 second per image crop. Finally, we evaluate our refiner on ModelNet and compare it with the state-of-the-art methods MP-AAE [16] and LatentFusion [15]. For this experiment, we remove the ShapeNet categories that overlap with the test ones in ModelNet from our training set in order to provide a fair comparison on novel instances and novel categories similar to [15, 16, 20]. Results reported in Table 2 show that our refiner significantly outperforms existing approaches across all metrics.

Rendered views	Rendered normals	BOP5	ModelNet ADD(0.1d)
1	✓	52.0	83.3
2	✓	59.0	90.4
4	✓	**61.7**	**96.1**
4		59.1	83.1

(a)

Training objects	Num. objects	BOP5	ModelNet ADD(0.1d)
GSO+ShapeNet	10 + 100	47.9	28.7
GSO+ShapeNet	100 + 1000	49.3	80.3
GSO+ShapeNet	250 + 2000	56.9	82.0
GSO+ShapeNet	500 + 10000	59.3	89.3
GSO+ShapeNet	1000 + 20000	61.7	**96.1**
GSO	1000	62.2	95.7
BOP	132	**62.6**	93.4

(b)

Table 3: Ablation study. We study (a) using multiple rendered object views and normal maps as input to our RGB-only refiner model and (b) training the refiner on different variations of our large-scale synthetic dataset. Average recall is reported on BOP5 (mean of LM-O, T-LESS, TUD-L, IC-BIN and YCB-V) and ModelNet.

4.3 Ablations

In this section we perform ablations of our approach to validate our main contributions. Additional ablations are in the appendix. For these ablations, we consider the RGB-only refiner and re-train several models with different configurations of hyper-parameters and training data.

Encoding the anchor point and object shape. As discussed in Section 3.1 the refiner must have information about the anchor point $\mathcal{O}$ in order to generalize to novel objects. We accomplish this by using 4 rendered views pointing towards the anchor point, see Figure 2(b). Table 3(a) shows the performance of the refiner increases as we increase the number of views from 1 to 4, validating our design choice. Multiple views may also help the network to understand the object's appearance from alternate viewpoints, thus potentially helping the refiner to overcome large initial pose errors. We also validate our choice to provide a normal map of the object to the network. This information can help the network use subtle object appearance variations that are only visible under different illumination like the details on the cross of a guitar.

Number of training objects. We now show the crucial role of the training data. We report in Table 3 (b) the results for our refiner trained on an increasing number of CAD models. The performance steadily increases with the number of objects, which validates that training on a large number of object models is important to generalize to novel ones. These results also suggest that the performance of our approach *could* be improved as more datasets of high-quality CAD models like GSO [7] become available.

Variety in the training objects. Next, we restrict the training to different sets of objects. We observe in the bottom of Table 3(b) that models from the GoogleScannedObjects are more important to the performance of the method on the BOP dataset compared to using both ShapeNet and GSO. We hypothesize this is due to the presence of high-quality textured objects in the GSO dataset. Finally, we train our model on the 132 objects of the BOP dataset. When testing on the same BOP objects, the performance benefits from knowing these objects during training is small compared to using our GSO+ShapeNet or GSO dataset.

4.4 Limitations

While MegaPose shows promising results in robot experiments (**please see the supplementary video**) and 6D pose estimation benchmarks, there is still room for improvement. We illustrate the failure modes of our approach in the supplementary material. The most common failure mode is due to inaccurate initial pose estimates from the coarse model. The refiner model is trained to correct poses that are within a constrained range but can fail if the initial error is too large. There exist multiple potential approaches to alleviate this problem. We can increase the number of pose hypotheses M at the expense of increased inference time, improve the accuracy of the coarse model, and increase the basin of attraction for refinement model. Another limitation is the runtime of our coarse model. We use $M = 520$ pose hypotheses per object which takes around 2.5 seconds to be rendered and

evaluated by our coarse model. In a tracking scenario however, the coarse model is run just once at the initial frame and the object can be tracked using the refiner which runs at 20Hz. Additionally, our refiner could also be coupled with alternate coarse estimation approaches such as [17, 18] to achieve improved runtime performance.

5 Conclusion

We propose MegaPose, a method for 6D pose estimation of novel objects. Megapose can estimate the 6D pose of novel objects given a CAD model of the object available only at test time. We quantitatively evaluated MegaPose on hundreds of different objects depicted in cluttered scenes, and performed ablation studies to validate our network design choices and highlight the importance of the training data. We release our models and large-scale synthetic dataset to stimulate the development of novel methods that are practical to use in the context of robotic manipulation where rapid deployment to new scenes with new objects is crucial. While this work focuses on the coarse estimation and fine refinement of an object pose, detecting any unknown object given only a CAD model is still a difficult problem that remains to be solved for having a complete framework for detection and pose estimation of novel objects. Future work will address zero-shot object detection using our large-scale synthetic dataset.

Acknowledgements

This work was partially supported by the HPC resources from GENCI-IDRIS (Grant 011011181R2), the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15 003/0000468), EU Horizon Europe Programme under the project AGIMUS (No. 101070165), Louis Vuitton ENS Chair on Artificial Intelligence, and the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

References

[1] Project page: https://megapose6d.github.io.

[2] T. Hodaň, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas. BOP Challenge 2020 on 6D object localization. In *ECCVW*, 2020.

[3] R. L. Haugaard and A. G. Buch. Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. In *CVPR*, 2022.

[4] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic. CosyPose: Consistent multi-view multi-object 6D pose estimation. In *ECCV*, 2020.

[5] L. Lipson, Z. Teed, A. Goyal, and J. Deng. Coupled iterative refinement for 6D multi-object pose estimation. In *CVPR*, 2022.

[6] G. Wang, F. Manhardt, F. Tombari, and X. Ji. GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation. In *CVPR*, 2021.

[7] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3D scanned household items. In *ICRA*, 2022.

[8] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield. Single-stage keypoint-based category-level object pose estimation from an RGB image. In *ICRA*, 2022.

[9] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, 2019.

[10] X. Chen, Z. Dong, J. Song, A. Geiger, and O. Hilliges. Category level object pose estimation via neural analysis-by-synthesis. In *ECCV*, 2020.

[11] F. Li, I. Shugurov, B. Busam, M. Li, S. Yang, and S. Ilic. Polarmesh: A star-convex 3d shape approximation for object pose estimation. *IEEE Robotics and Automation Letters*, 7(2):4416–4423, 2022.

[12] F. Manhardt, G. Wang, B. Busam, M. Nickel, S. Meier, L. Minciullo, X. Ji, and N. Navab. CPS++: Improving class-level 6D pose and shape estimation from monocular images with self-supervised learning. *arXiv preprint arXiv:2003.05848*, 2020.

[13] L. Manuelli, W. Gao, P. Florence, and R. Tedrake. KPAM: Keypoint affordances for category-level robotic manipulation. In *ISRR*, 2019.

[14] B. Wen, W. Lian, K. Bekris, and S. Schaal. Catgrasp: Learning category-level task-relevant grasping in clutter from simulation. *arXiv preprint arXiv:2109.09163*, 2021.

[15] 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.

[16] 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.

[17] I. Shugurov, F. Li, B. Busam, and S. Ilic. OSOP: A multi-stage one shot object pose estimation framework. In *CVPR*, 2022.

[18] V. N. Nguyen, Y. Hu, Y. Xiao, M. Salzmann, and V. Lepetit. Templates for 3D object pose estimation revisited: Generalization to new objects and robustness to occlusions. In *CVPR*, 2022.

[19] Y. Xiao, X. Qiu, P.-A. Langlois, M. Aubry, and R. Marlet. Pose from shape: Deep pose estimation for arbitrary 3D objects. In *BMVC*, 2019.

[20] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. DeepIM: Deep iterative matching for 6D pose estimation. In *ECCV*, 2018.

[21] B. Okorn, Q. Gu, M. Hebert, and D. Held. ZePHyR: Zero-shot pose hypothesis rating. In *ICRA*, 2021.

[22] Y. Liu, Y. Wen, S. Peng, C. Lin, X. Long, T. Komura, and W. Wang. Gen6D: Generalizable model-free 6-DoF object pose estimation from RGB images. In *arXiv:2204.10776*, 2022.

[23] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*, 2010.

[24] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(n) solution to the PnP problem. *IJCV*, 2009.

[25] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.

[26] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13(2):119–152, 1994.

[27] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. In *ICCV*, 2017.

[28] A. Osokin, D. Sumin, and V. Lomakin. Os2d: One-stage one-shot object detection by matching anchor features. In *ECCV*. Springer, 2020.

[29] J.-P. Mercier, M. Garon, P. Giguere, and J.-F. Lalonde. Deep template-based object instance detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1507–1516, 2021.

[30] Y. Xiao and R. Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *European Conference on Computer Vision (ECCV)*, 2020.

[31] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019.

[32] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[33] L. G. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[34] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artif. Intell.*, 1987.

[35] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[36] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006.

[37] A. Collet and S. S. Srinivasa. Efficient multi-view object recognition and full pose estimation. In *ICRA*, 2010.

[38] A. Collet, M. Martinez, and S. S. Srinivasa. The MOPED framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, 2011.

[39] 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.

[40] F. Jurie and M. Dhome. A simple and efficient template matching algorithm. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 544–549. IEEE, 2001.

[41] M. Rad and V. Lepetit. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. In *CVPR*, 2017.

[42] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. In *Conference on Robot Learning (CoRL)*, 2018.

[43] B. Tekin, S. N. Sinha, and P. Fua. Real-time seamless single shot 6d object pose prediction. In *CVPR*, 2018.

[44] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019.

[45] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis. 6-dof object pose from semantic keypoints. In *ICRA*, 2017.

[46] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann. Segmentation-driven 6D object pose estimation. In *CVPR*, 2019.

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

[48] K. Park, T. Patten, and M. Vincze. Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation. In *ICCV*, 2019.

[49] C. Song, J. Song, and Q. Huang. Hybridpose: 6D object pose estimation under hybrid representations. In *CVPR*, 2020.

[50] S. Zakharov, I. Shugurov, and S. Ilic. Dpod: 6d pose object detector and refiner. In *CVPR*, 2019.

[51] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[52] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352, 2019.

[53] K. Pauwels and D. Kragic. Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1300–1307. IEEE, 2015.

[54] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *ICCV*, 2015.

[55] F. Manhardt, W. Kehl, N. Navab, and F. Tombari. Deep model-based 6d pose refinement in rgb. In *ECCV*, 2018.

[56] M. Aubry, D. Maturana, A. A. Efros, B. Russell, and J. Sivic. Seeing 3D chairs: Exemplar part-based 2D-3D alignment using a large dataset of CAD models. In *CVPR*, 2014.

[57] Y. Xiao, Y. Du, and R. Marlet. PoseContrast: Class-agnostic object viewpoint estimation in the wild with pose-aware contrastive learning. In *3DV*, 2021.

[58] Y. He, Y. Wang, H. Fan, J. Sun, and Q. Chen. FS6D: Few-shot 6D pose estimation of novel objects. In *CVPR*, 2022.

[59] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. Zhou. OnePose: One-shot object pose estimation without CAD models. In *CVPR*, 2022.

[60] E. Olson. AprilTag: A robust and flexible visual fiducial system. In *ICRA*, 2011.

[61] 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. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[62] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *ECCV*, 2018.

[63] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian conference on computer vision (ACCV)*, 2012.

[64] T. Hodan, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of Texture-Less objects. In *Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[65] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *CVPR*, 2016.

[66] B. Drost, M. Ulrich, P. Bergmann, P. Hartinger, and C. Steger. Introducing mvtec itodd-a dataset for 3d object recognition in industry. In *ICCV Workshops*, 2017.

[67] R. Kaskman, S. Zakharov, I. Shugurov, and S. Ilic. Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. In *ICCV Workshops*, 2019.

[68] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019.

[69] The pillow imaging library. https://github.com/python-pillow/pillow.

[70] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg. Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. In *2018 IEEE International Conference on robotics and automation (ICRA)*, pages 5620–5627. IEEE, 2018.

[71] C. Xie, Y. Xiang, A. Mousavian, and D. Fox. Unseen object instance segmentation for robotic environments. *IEEE Transactions on Robotics*, 37(5):1343–1359, 2021.

[72] B. Wen, C. Mitash, B. Ren, and K. E. Bekris. se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10367–10373. IEEE, 2020.

Appendix

This appendix is organized as follows. In section A, we provide the equations of the pose update predicted by the refiner network, and show it depends on the anchor point. In section B, we give details on the loss used to train the refiner network. Section C explains the normalization strategy we apply to the observed and rendered depth images of the RGB-D refiner. Section D details the pose hypotheses used during training and inference of the coarse model. In section E, we provide examples of training images and give details on the data augmentation and training hardware. In section F, we perform additional ablations to validate (i) the contributions of our coarse network, (ii) the choice of hyper-parameter M. We also provide details on the robot experiments shown in the supplementary video. In section G, we illustrate qualitatively that our approach is robust to illumination condition variations. Section H illustrates the main failure modes of our approach. Finally, section I investigates the robustness of our approach with respect to an incorrect 3D model.

The **supplementary video** shows predictions of our approach on real images. We apply our approach in *tracking mode* on several videos. Tracking consists in running the coarse estimator on the first frame of a video sequence, and then applying one iteration of the refiner on each new image, using the prediction in the previous image as the pose initialization at the input of refinement network. This approach can process 20 images per second. The video notably demonstrates the method is robust to occlusion and can be used to perform visually guided robotic manipulation of novel objects.

A Pose update and anchor point

Pose update. We use the same pose update as DeepIM [20] and CosyPose [4]. The network predicts 9 values corresponding to one 3-vector $[v_x, v_y, v_z]$ to predict an update of the translation of a 3D anchor point, and two 3-vectors e_1, e_2 that define a rotation update explained below. The pose update consists in updating (i) the position of a 3D reference point $\mathcal{O}$ attached on the object, and (ii) the rotation matrix R_{CO} of the object frame expressed in the camera frame (please note the different notations for the anchor point $\mathcal{O}$ and the object frame O):

$$x_{\mathcal{O}}^{k+1} = \left(\frac{v_x}{f_x^C} + \frac{x_{\mathcal{O}}^k}{z_{\mathcal{O}}^k}\right) z_{\mathcal{O}}^{k+1}, \tag{1}$$

$$y_{\mathcal{O}}^{k+1} = \left(\frac{v_y}{f_y^C} + \frac{y_{\mathcal{O}}^k}{z_{\mathcal{O}}^k}\right) z_{\mathcal{O}}^{k+1}, \tag{2}$$

$$z_{\mathcal{O}}^{k+1} = v_z z_{\mathcal{O}}^k, \tag{3}$$

$$R_{CO}^{k+1} = R(e_1, e_2) R_{CO}^k, \tag{4}$$

where $[x_{\mathcal{O}}^k, y_{\mathcal{O}}^k, z_{\mathcal{O}}^k]$ is the 3D position of the anchor point expressed in camera frame at iteration k, R_{CO}^k a rotation matrix describing the objects orientation expressed in camera frame, f_x^C and f_y^C are the (known) focal lengths that correspond to the (virtual) camera associated with the cropped observed image, and $R(e_1, e_2)$ is a rotation matrix describing the rotation update recovered from e_1, e_2 using [68] by orthogonalizing the basis defined by the two predicted rotation vectors e_1, e_2 similar to [4]. Finally, $[x_{\mathcal{O}}^{k+1}, y_{\mathcal{O}}^{k+1}, z_{\mathcal{O}}^{k+1}]$ and R_{CO}^{k+1} are, respectively, the translation and rotation after applying the pose update. The 3D translation of the anchor point and the rotation matrix R_{CO} are used to define the pose the object.

Dependency to the anchor point. We now show that the predictions the network must make to correct a pose error between an initial pose $\mathcal{T}_{CO}^k$ and a target pose $\mathcal{T}_{CO}^{k+1}$ is independent of the choice of the orientation of the objects coordinate frame O but depends on the choice anchor point $\mathcal{O}$. Let us denote $\mathcal{O}^1, \mathcal{O}^2$ two different anchor points, and R_{CO^1}, R_{CO^2} the rotation matrices of the object (expressed in the fixed camera frame) for two different choices of object coordinate frames O^1 and O^2. We note $t_{\mathcal{O}^1\mathcal{O}^2} = \mathcal{O}^2 - \mathcal{O}^1 = [x_{12}, y_{12}, z_{12}]$ the 3D translation vector between $\mathcal{O}^2$ and $\mathcal{O}^1$; and $R_{O^1O^2} = R_{CO^1}^T R_{CO^2}$ the rotation of coordinate frame O^2 expressed in O^1. For one choice of anchor point and object frame, e.g. $\mathcal{O}_1$ and R_{CO^1}, we derive the predictions the network has to make to correct the error using equations (1),(2),(3),(4):

$$v_x^1 = f_x^c \left(\frac{x_{O^1}^{k+1}}{z_{O^1}^{k+1}} - \frac{x_{O^1}^k}{z_{O^1}^k}\right) \tag{5}$$

$$v_y^1 = f_y^C \left(\frac{y_{O^1}^{k+1}}{z_{O^1}^{k+1}} - \frac{y_{O^k}^k}{z_{O^k}^k}\right) \tag{6}$$

$$v_z^1 = \frac{z_{O^1}^{k+1}}{z_{O^1}^k} \tag{7}$$

$$R^1 = R_{CO^1}^{k+1} \left(R_{CO^1}^k\right)^T, \tag{8}$$

and similar for 2 by replacing the superscript. From these equations, we derive:

$$v_x^1 - v_x^2 = f_x^c \left(\frac{x_{O^1}^{k+1}}{z_{O^1}^{k+1}} - \frac{x_{O^1}^k}{z_{O^1}^k} - \frac{x_{O^2}^{k+1}}{z_{O^2}^{k+1}} + \frac{x_{O^2}^k}{z_{O^2}^k}\right) \tag{9}$$

$$v_y^1 - v_y^2 = f_y^C \left(\frac{y_{O^1}^{k+1}}{z_{O^1}^{k+1}} - \frac{y_{O^1}^k}{z_{O^1}^k} - \frac{y_{O^2}^{k+1}}{z_{O^2}^{k+1}} + \frac{y_{O^2}^k}{z_{O^2}^k}\right) \tag{10}$$

$$v_z^1 - v_z^2 = \frac{z_{O^1}^{k+1}}{z_{O^1}^k} - \frac{z_{O^2}^{k+1}}{z_{O^2}^k} \tag{11}$$

$$R^1 \left(R^2\right)^T = R_{CO^1}^{k+1} \left(R_{CO^1}^k\right)^T R_{CO^2}^k \left(R_{CO^2}^{k+1}\right)^T = R_{CO^1}^{k+1} R_{O^1O^2} R_{O^2C}^{k+1} = Id. \tag{12}$$

From eq. (12), we have $R^1 \left(R^2\right)^T = Id$. In other words, the rotation matrices that the network must predict to correct the errors in scenarios 1 and 2 are the same. The network predictions for the rotation components thus do not depend on the choice of the choice of object coordinate system. However the other components of the translation cannot be simplified further. For example, derivations of eq. (11) leads to $v_z^1 - v_z^2 = \frac{z_{12}\left(z_{\mathcal{O}1}^{k+1} - z_{\mathcal{O}1}^{k}\right)}{z_1^k(z_1^k + z_{12})}$ which is non-zero in the general case where O^1 and O^2 are different and there is an error between the initial and target poses. This proves that different choices of anchor point leads to different predictions. For the network to generalize to a novel object, the network be able to infer the 3D position of the anchor point on this object. We achieve this by rendering multiple views of the objects in which the anchor point reprojects to the center of each image as explained in Section 3.1 of the main paper.

B Refiner loss

Our refiner network is trained using the same loss as in CosyPose [4], but without using symmetry information on the objects because it is not typically not available for large-scale datasets of CAD models like ShapeNet or GoogleScannedObjects. We first define the distance $D_O(\mathcal{T}_1, \mathcal{T}_2)$ to measure the distance between two 6D poses represented by transformations $\mathcal{T}_1$ and $\mathcal{T}_2$ using the 3D points $\mathcal{X}_O$ of an object O:

$$D_O(\mathcal{T}_1, \mathcal{T}_2) = \frac{1}{|\mathcal{X}_O|} \sum_{x \in \mathcal{X}_O} |\mathcal{T}_1 x - \mathcal{T}_2 x|, \tag{13}$$

where $|\cdot|$ is the L_1 norm. In practice, we uniformly sample 2000 points on the surface of an object's CAD model to compute this distance. We also define the pose update function F which takes as input the initial estimate of the pose $\mathcal{T}_{CO}^k$, the predictions of the neural network $[v_x, v_y, v_z]$ and R, and outputs the updated pose:

$$\mathcal{T}_{CO}^{k+1} = F(\mathcal{T}_{CO}^k, [v_x, v_y, v_z], R), \tag{14}$$

where the closed form of function F is expressed in equations (1) (2) (3) (4). We also write $[v_x^\star, v_y^\star, v_z^\star]$ and $R^\star$ as the target predictions, i.e. the predictions such that $\mathcal{T}_{CO}^\star = F(\mathcal{T}_{CO}^k, [v_x^\star, v_y^\star, v_z^\star], R^\star)$, where $\mathcal{T}_{CO}^\star$ is the ground truth camera-object pose. The loss used to train the refiner is the following:

$$\mathcal{L} = \sum_{k=1}^{K} D_O(F(\mathcal{T}_{CO}^k, [v_x, v_y, v_z^\star], R^\star), \mathcal{T}_{CO}^\star) \tag{15}$$

$$+ D_O(F(\mathcal{T}_{CO}^k, [v_x^\star, v_y^\star, v_z], R^\star), T_{CO}^\star) \tag{16}$$

$$+ D_O(F(\mathcal{T}_{CO}^k, [v_x^\star, v_y^\star, v_z^\star], R), \mathcal{T}_{CO}^\star), \tag{17}$$

where D_O is the distance defined in eq. (13) and K is the number of training iterations. The different terms of this loss separate the influence of: xy translation (15), relative depth (16) and rotation (17). We sum the loss over $K = 3$ refinement iterations to imitate how the refinement algorithm is applied at test time but the error gradients are not backpropagated through rendering and iterations. For simplicity, we write the loss for a single training sample (i.e. a single object in an image), but we sum it over all the samples in the training set.

C Depth normalization

When depth measurements are available, the observed depth image and depth images of the renderings are concatenated with the images, as mentioned in Section 3.1 of the main paper. At test time, the objects may be observed at different depth outside of the training distribution. In order for the network to become invariant to the absolute depth values of the inputs, we normalize both observed and rendered depth. Let us denote D a depth image (rendered or observed are treated similarly). We apply the following operations to D. (i) Clipping of the metric depth values of D to lie between 0 and $z_{\mathcal{O}}^k + 1$, where $z_{\mathcal{O}}^k$ is the depth of the anchor point on the object in the input pose at iteration k:

$$D \leftarrow \text{clip}(D^k, 0, z_{\mathcal{O}}^k + 1), \tag{18}$$

and (ii) centering of the depth values:

$$D \leftarrow \frac{D}{z^k_{\mathcal{O}}} - 1. \tag{19}$$

D Pose hypotheses in the coarse model

Training hypotheses. Given the ground truth object pose $\mathcal{T}^\star_{CO}$, we generate a perturbed pose $\mathcal{T}'_{CO}$ by applying random translation and rotation to $\mathcal{T}'_{CO}$. The parameters of this (small) perturbation are sampled from the same distribution as the distribution used to sample the perturbed poses the refiner network is trained to correct. The translation is sampled from a normal distribution with a standard deviations of $(0.02, 0.02, 0.05)$ centimeters and rotation is sampled as random Euler angles with a standard deviation of 15 degrees in each axis.

We then define several poses that depend on $\mathcal{T}'_{CO}$ and cover a large variety of viewing angles of the object. We define a cube of size $2z'_{\mathcal{O}}$, where $z'_{\mathcal{O}}$ is the z component of the 3D translation in the pose $\mathcal{T}'_{CO}$. The CAD model of the object observed under orientation R'_{CO} is placed at the center of the cube. We then place 26 cameras at the locations of each corner, half-side and face centers of the cube. By construction, one of these cameras, which we denote C^0, has the same camera-object orientation as $\mathcal{T}'_{CO}$, and all others $\{C^i\}_{i=1..25}$ correspond to cameras observing the object under viewpoints which are sufficiently far from R_{C^0O} and outside the basin of attraction of the refiner by construction. In addition, we apply inplane rotations of $90°$, $180°$ and $270°$ to each camera, which leads to a total of $26 * 4 = 104$ cameras with one positive and 103 negatives.

We mark $\mathcal{T}_{C^0O}$ as a *positive* for the coarse model because the error between $\mathcal{T}_{C^0O}$ and $\mathcal{T}^\star_{CO}$ lies within the basin of attraction of the refiner. All other cameras are marked as negatives. During training, the positives account for around 30% of the total numbers of images in a mini-batch.

Test hypotheses. At test time, a 2D detection of the object is available. Let $u_{det} = (u_{det,x}, u_{det,y})$ and $(\Delta u_{det} = \Delta u_{det,x}, \Delta u_{det,y})$ define the center and the size of the approximate 2D bounding box of the object in the image. We start by defining a random camera-object orientation R^p. The anchor point on the object is set to match the center of the bounding box u_{det}. We make a first hypothesis of the depth of the anchor by setting $z^{guess}_{\mathcal{O}^p} = 1\text{m}$ and use this initial value to estimate the coordinates $x_{\mathcal{O}^p}$ and $y_{\mathcal{O}^p}$ of the anchor point in the camera frame:

$$x^{guess}_{\mathcal{O}^p} = u_{det,x} \frac{z^{guess}_{\mathcal{O}^p}}{f_x} \tag{20}$$

$$y^{guess}_{\mathcal{O}^p} = u_{det,y} \frac{z^{guess}_{\mathcal{O}^p}}{f_y}, \tag{21}$$

where f_x and f_y are the (known) focal lengths of the camera. We then update the depth estimate $z^{guess}_{\mathcal{O}^p}$ using the following simple strategy. We project the points of the object 3D model using R^p and the initial guess of the 3D position of the anchor point we have just defined. These points define a bounding box with dimensions $\Delta u_{guess,x} = (\Delta u_{guess,x}, \Delta u_{guess,y})$ and the center remains unchanged $u_{guess} = u_{det}$ by construction. We compute an updated depth of the anchor point such that its width and height approximately match the size of the 2D detection:

$$z_{\mathcal{O}^p} = z^{guess}_{\mathcal{O}^p} \frac{1}{2} \left(f_x \frac{\Delta u_{guess,x}}{\Delta u_{det,x}} + f_y \frac{\Delta u_{guess,y}}{\Delta u_{det,y}} \right) \tag{22}$$

and use this new depth to compute $x_{\mathcal{O}^p}$ and $y_{\mathcal{O}^p}$ using equations (20) and (21) that were used to define $x^{guess}_{\mathcal{O}^p}$ and $y^{guess}_{\mathcal{O}^p}$. The rotation R^p and 3-vector $[x_{\mathcal{O}^p}, y_{\mathcal{O}^p}, z_{\mathcal{O}^p}]$ define the pose of hypothesis p. We then use the same strategy used to define the training hypotheses (described above) in order to define 103 additional viewpoints depending on p. We repeat the operation $P = 5$ times, for a total of $5 * 104 = 520$ pose hypotheses.

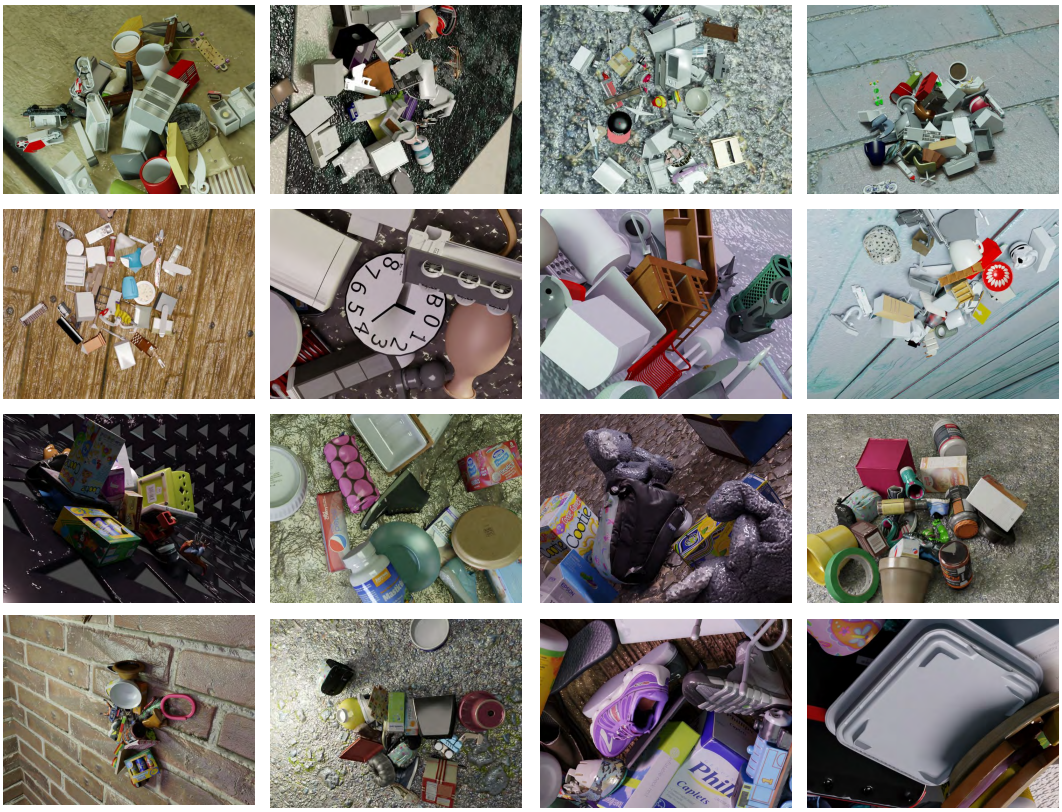

Figure 4: Example of training images. Randomly sampled images from our large-scale synthetic dataset generated with BlenderProc [31]. CAD models from the ShapeNet [61] and GoogleScannedObjects [7] datasets are used.

Pose hypotheses	YCB-V	LM-O
PPF	52.7	34.4
PPF+Zephyr [21]	59.8	45.8
PPF+Our coarse	**61.6**	**52.1**

Table 4: Performance of the coarse model. We compare the performance of our coarse network with Zephyr [21].

E Training details

Training images. We generate 2 million photorealistic images using BlenderProc [31] as explained in Section 3.2 of the main paper. Randomly sampled images from the training set are shown in Figure 4.

Data augmentation. We apply data augmentation to the synthetic images during training. We use the same data augmentation as CosyPose [4] for the RGB images. It includes Gaussian blur, contrast, brightness, colors and shaprness filters from the Pillow library [69]. For the depth images, we take inspiration from the augmentations used in [70, 71, 72]. Augmentations include blur, ellipse dropout, correlated and uncorrelated noise.

GPU hardware and training time. Training time is respectively 32 and 48 hours for the coarse and refiner models using 32 V-100 GPUs. This training is performed once, and estimating the pose of novel objects does not require any fine-tuning on the target objects.

F Additional experiments.

Coarse network. In order to evaluate the validate the contributions of our coarse scoring network, we use a set of pose hypotheses generated for novel objects by the commercial Halcon 20.05 Progress software which implements the PPF algorithm described in [22]. Note that these are the same pose hypotheses used in Zephyr [21]. We then find the best hypotheses using the scores of PPF, the scoring network of Zephyr or our coarse network, and report AR results for the LM-O and YCB-V datasets in the table 4. On both datasets, our coarse network is better than the two baselines (PPF and Zephyr) for selecting the best poses among a given set of hypotheses.

Classification-based coarse network. To validate our classification-based coarse model, we consider a regression-based alternative. We trained a regression-based network similar to the coarse model of CosyPose [4] which takes as input six views of the objects covering viewpoints at the poles of a sphere centered on the object. The network collapsed during training, leading to large errors that cannot be recovered by the refiner and a performance close to zero on the BOP datasets. We hypothesize this failure is due to the presence of symmetric objects in our training set which leads to ambiguous gradients during training. This failure could also be attributed to other factors, such as the difficulty to interpret the full 3D geometry of an object with a CNN given six views of its 3D model captured under distant viewpoints.

Number of coarse pose hypotheses. M is an important parameter of our method, which can be used to choose a trade-off between running time and accuracy. The performance significantly improves from M=104 to M=520 (+11.4 AR on BOP5) while keeping the running-time of the coarse model reasonable (1.6 seconds for M=520 compared to 0.3 seconds for M=120). Above M=520, the performance improvement is marginal, e.g. (+0.9 AR) for 4608 hypotheses. Please note that we are still making improvements to our code and have lower runtimes than reported in the paper (1.6s for M=520 compared to the 4s mentioned in line 276).

Robotic grasping experiments. We performed a qualitative real-robot grasping experiment. For multiple YCB-V objects, we manually annotated one grasp with respect to the object's coordinate frame. We then placed the considered object (e.g. the drill in the supplementary video) in a scene among other objects representing visual distractors. The object may be placed on the table or on another object. We then take a single RGB image of the scene using a RealSense D415 camera mounted on the gripper of a Franka Emika Panda robot. We detect the object in 2D using the Mask-RCNN detector from CosyPose [4], and run our Megapose approach composed of coarse and refiner modules for estimating the 6D pose of the object with respect to the camera. We then express the 6D pose of the object and grasp with respect to the robot using the known camera-to-robot extrinsic calibration. We then use a motion planner to generate a robot motion that reaches the estimated grasp pose with the gripper and lift the object. This experiment shown in the supplementary video shows that the pose estimates are of sufficiently high quality to be useful for a robotic manipulation task.

G Robustness to illumination conditions

In Figure 5, we show qualitative predictions of our approach for the watering can on the TUD-L dataset. Please notice the high accuracy of our approach despite challenging illumination conditions.

H Failure modes and performance on specific types of objects

We carry out a per-object analysis of the performance of our approach on the YCB-V dataset. For each of the 21 objects of the dataset, we report the percentage of predictions for which the error with the ground truth is within a threshold of 15° in rotation and 5cm in translation. Results are reported in Figure 6.

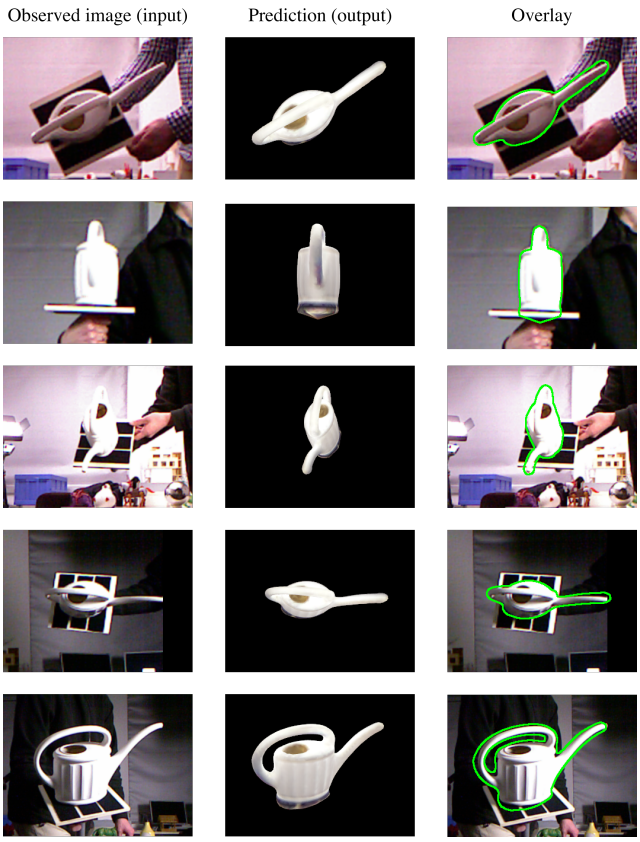

Figure 5: Qualitative examples on the TUD-L dataset. Each row presents one example prediction on a real image. The first column is the real observed image, the second column is the prediction of our approach here illustrated using a rendering of the object's CAD model in the predicted pose, and an overlay of the prediction and output is shown on the right.

Next, we illustrate the main failure modes of our approach using a set of objects which have a performance below average on this dataset. Examples of failure cases are presented in Figure 7. We observed three main failure modes to our approach. First, we observe the orientation of a novel object may be incorrectly predicted if the object has a similar visual appearance under different viewpoint. We observed this failure mode in particular for textureless objects such as a red bowl that appears similar whether it is standing upside or it is flipped. Second, we observe that our approach may fail to disambiguate the pose of objects that are asymmetric but for which it is necessary to look at fine details on the objects to disambiguate multiple possible poses. An example is a pair of scissors which have left and right handles with slightly different dimensions. In both of these failure modes, we observed that our refiner gets stuck into a local minimal due to an inaccurate coarse estimate outside of the basin of attraction of our refiner model. Finally, using a CAD model with incorrect scale leads to an incorrect estimation of the depth of the object due to the object scale/depth ambiguity in RGB images. We observe for example that the translation estimates of the wooden block of YCB-V have systematically large error despite the rendering of our prediction correctly matching the contours of the object in the observed image. This is because the scale of the CAD model of the wooden block publicly available does not match the correct dimensions of the real object which was used for annotating the ground truth.

I 3D model quality

Our approach can be applied even if the 3D model of the object does not exactly matches the real object. In figure 8, we show examples of correctly estimated poses using low-fidelity CAD models with low-quality textures or geometric discrepancies between the real object and its 3D model.

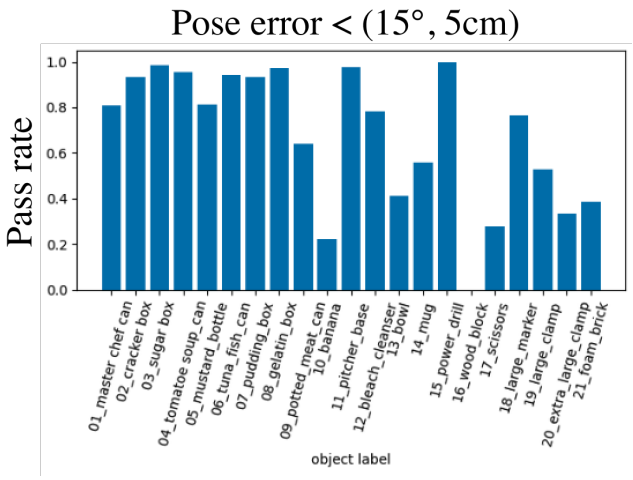

Figure 6: Per-object analysis on the YCB-V dataset. For each object, we report the percentage of estimates for which the error between our pose prediction and the ground truth is within 5 centimeters in translation and 15 degrees in rotation.

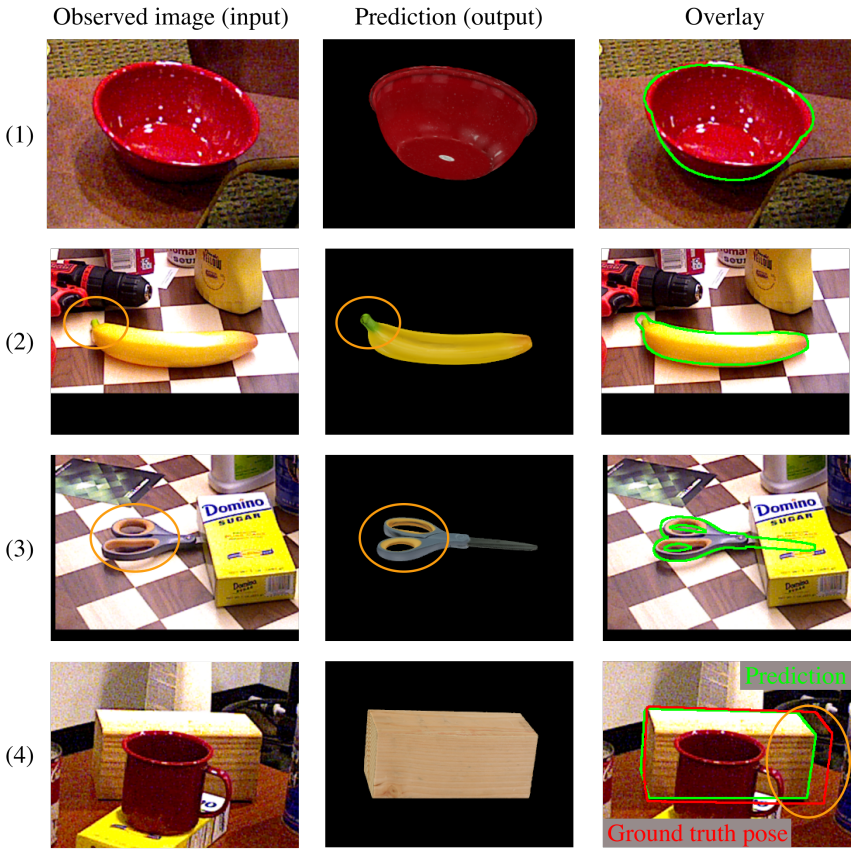

Figure 7: Illustration of the main failure modes of our approach. In (1) and (2), the contours of the object in the predicted poses correctly overlay the observed image, but the pose is incorrect because these objects have a similar appearance under different viewpoints. In (3), our approach fails to correctly distinguish the left and right handles with different dimensions in order to disambiguate the orientation of the asymmetric pair of scissors. In (4), our pose prediction does not match the ground truth annotation, because the CAD model of the wooden block we use for pose estimation has different dimensions that do not match the dimensions of the real objects which was used for annotating the ground truth. Please notice in all examples how the contours of the object in the predicted pose are closely aligned with the contours of the object in the input image.

Figure 8: Predictions using low-fidelity CAD models. In (a) we show the result of our approach on LineMOD Occlusion for three different objects which have only low-fidelity CAD models available. In (1) and (2), the quality of the mesh and textures is poor as illustrated in (b). Notice for example how the annotations on the glue box or the brand of the drill are not readable on the CAD models. In (3), the hole of the watering can does not appear in the CAD model. Despite these discrepancies between the real object and the CAD model, our approach correctly estimates the pose of each object.