FreeZe: Training-free zero-shot 6D pose estimation with geometric and vision foundation models

Andrea Caraffa[1], Davide Boscaini[1], Amir Hamza[1,2], and Fabio Poiesi[1]

[1] Fondazione Bruno Kessler, Trento, Italy
{acaraffa,dboscaini,ahamza,poiesi}fbk.eu
[2] University of Trento, Italy

Abstract. Estimating the 6D pose of objects unseen during training is highly desirable yet challenging. Zero-shot object 6D pose estimation methods address this challenge by leveraging additional task-specific supervision provided by large-scale, photo-realistic synthetic datasets. However, their performance heavily depends on the quality and diversity of rendered data and they require extensive training. In this work, we show how to tackle the same task but without training on specific data. We propose FreeZe, a novel solution that harnesses the capabilities of pre-trained geometric and vision foundation models. FreeZe leverages 3D geometric descriptors learned from unrelated 3D point clouds and 2D visual features learned from web-scale 2D images to generate discriminative 3D point-level descriptors. We then estimate the 6D pose of unseen objects by 3D registration based on RANSAC. We also introduce a novel algorithm to solve ambiguous cases due to geometrically symmetric objects that is based on visual features. We comprehensively evaluate FreeZe across the seven core datasets of the BOP Benchmark, which include over a hundred 3D objects and 20,000 images captured in various scenarios. FreeZe consistently outperforms all state-of-the-art approaches, including competitors extensively trained on synthetic 6D pose estimation data. Code will be publicly available at andreacaraffa.github.io/freeze.

Keywords: Object 6D pose estimation · Geometric foundation models · Vision foundation models · Zero-shot learning

1 Introduction

In our daily interactions, we easily manipulate objects around us, whether by grasping a mug or pouring water into a glass, thanks to our subconscious ability to locate them in the real world. In the realm of machine vision, this task is formalized as object 6D pose estimation, which involves determining the rotation and translation of an object within a scene, relative to a global reference frame. This becomes the key for applications such as robotic manipulation [33, 59], augmented reality [48], and autonomous driving [22].

There exist different categories of object 6D pose estimation methods, depending on the available information about the object of interest. The first

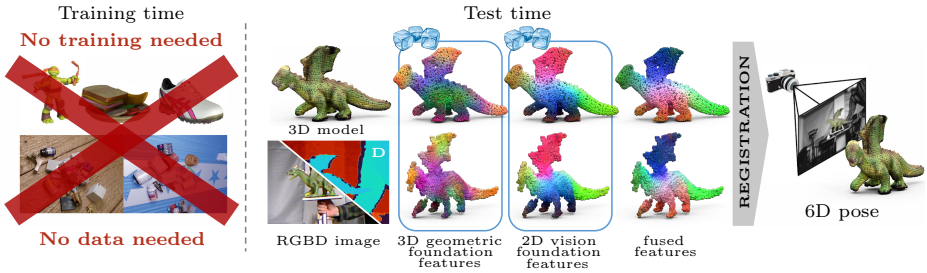

Fig. 1: FreeZe is designed for zero-shot object 6D pose estimation, which involves estimating the pose of an unseen 3D object within a 3D scene (*e.g.* posing the dragon 3D model within the RGBD image displayed in the center). Unlike most competitors, our approach eliminates the need for extensive training or the generation of large-scale tailored datasets (depicted on the left). Instead, FreeZe builds features suitable for 3D registration by fusing 3D geometric features with 3D-lifted 2D visual features extracted from separate pre-trained foundation models.

categorization is between *model-based* and *model-free* methods: the former require the object's 3D model as input [3, 6, 35, 51], while the latter relax this assumption and only require a set of reference views [13, 19] or a video [17, 49] of the input object. The second distinction lies among *instance-level*, *category-level*, and *zero-shot* methods. Instance-level methods [41, 45, 51] perform pose estimation for specific object instances seen during training (*e.g.* a particular type of drill) but do not generalize well to unseen objects. Category-level methods [16, 27, 32] generalize to different object instances within a category (*e.g.* different types of drills) but do not generalize well to unseen object categories. Zero-shot methods [3, 26, 40] address the generalization challenge by performing pose estimation for both unseen objects and categories. This capability facilitates real-world application deployment, eliminating the need to repeatedly retrain pose estimators for every new object encountered within the application's context.

Our work focuses on the model-based zero-shot setting. Within this context, most competitors use extensive training on large-scale pose estimation datasets, which are meticulously generated to meet specific criteria that fulfill the zero-shot paradigm. For example, MegaPose [26] utilizes physically-based rendering techniques to generate a synthetic dataset comprising two million images encompassing 20 thousand distinct objects. However, these methods face two main limitations: firstly, their performance heavily relies on the quality and diversity of the rendered data, and secondly, their training process demands considerable time and resources. Therefore, we aim to answer this fundamental question: "Do we really need task-specific training at the time of foundation models?". To answer No! to this question, we propose a novel training-**free ze**ro-shot approach, *FreeZe* for short, that harnesses the capabilities of pre-trained geometric and vision foundation models without requiring any training (Fig. 1). As geometric foundation model we leverage GeDi [43], which has been trained on 3D point cloud data of indoor scenes, while we use DINOv2 [39] as vision foundation

model, which has been trained on web-scale 2D images with self-supervision. FreeZe comprises four modules: *feature extraction*, *feature fusion*, *pose estimation*, and *pose refinement*. During feature extraction, we first compute 3D geometric features for the input object's 3D model using a frozen GeDi, and 2D visual features from the input image using a frozen DINOv2. We feed DINOv2 with 2D images rendered from different viewpoints around the object in order to compute 2D visual features for the object's 3D model. We feed GeDi with a 3D point cloud obtained by 3D-lifting the input depth in order to compute 3D geometric features for the scene. During feature fusion, we compute distinctive 3D point-level features by concatenating and regularizing geometric and visual features. The fused features are more discriminative because they encode both the geometric and visual characteristics, providing a richer and more comprehensive representation. During pose estimation, we use RANSAC to establish valid correspondences between the 3D points of the object and the 3D-lifted scene. In the case of geometrically-symmetric objects, we further refine their poses with a novel symmetry-aware refinement procedure based on DINOv2 visual features. We comprehensively evaluate FreeZe using the seven core datasets of the BOP Benchmark [1], which include LM-O [4], T-LESS [20], TUD-L [21], IC-BIN [11], ITODD [12], HB [24], and YCB-V [53]. Together, these datasets encompass over a hundred objects and 20 thousand images, spanning a wide range of scenarios (*e.g.* industrial vs ordinary environments) and types of noise (occlusions, variations in illumination, texture-less or symmetrical objects, cluttered scenes). FreeZe consistently outperforms all state-of-the-art approaches, including competitors extensively trained on large-scale synthetic pose estimation data. In summary, our contributions are:

- We are the first that effectively leverage the synergy between geometric and vision foundation models for the task of 6D pose estimation of unseen objects;
- We perform 6D pose estimation without requiring any task-specific training, resulting in a versatile solution that can be easily integrated with foundation models that may emerge in the future;
- We establish state-of-the-art performance on the BOP Benchmark, outperforming other competitors by a significant margin without requiring any additional training on task-specific data.

2 Related work

3D deep descriptors are typically used for generic point cloud registration problems [9, 43]. Corsetti et al. [10] showed that these descriptors can be adapted to the challenge of object 6D pose estimation. Deep descriptors can be based on the Local Reference Frame (LRF), which transforms a local set of points (patch) into a canonical representation to achieve rotation and translation invariance [54]. Canonicalized points are then processed by a neural network to produce a compact descriptor as output. Methods in this category include G3DOA [55], which utilizes multi-scale cylindrical convolutions; WSDesc [28], based on voxelization layers that learn optimal voxel neighborhood size; Li et al. [29], which employs

multi-view differentiable rendering; and GeDi [43], which processes canonicalized points through a PointNet++ [44] network. Descriptors can also be computed without relying on LRF. Methods in this category include SpinNet [2], which processes points projected on a cylindrical kernel using 3D cylindrical convolutions; FCGF [9], based on sparse convolutions for learning point-level 3D descriptors; and PREDATOR [23], which uses an attention mechanism to handle low-overlap point clouds. These descriptors are tailored for registering point clouds of similar types, dealing with structures of points that differ significantly from those found in object 6D pose estimation benchmarks. In our work, we have successfully employed these types of descriptors to establish new state-of-the-art results in object 6D pose estimation.

2D vision foundation models, including CLIP [46], DINOv2 [39], ImageBind [14], and Segment Anything Model (SAM) [25], can be employed across a range of tasks and domains. CLIP [46] connects visual concepts with textual descriptions using a ViT as image encoder and a GPT-like transformer as textual encoder, respectively. DINOv2 [39] adopts self-supervised training on large-scale unlabeled data, introducing a patch-level objective to capture fine-grained image details. ImageBind [14] integrates data from six different modalities into a shared feature space. This integration is achieved with image-paired data, thereby eliminating the need for combinations of all modalities. SAM [25] is specifically designed for object segmentation based on visual prompts such as a single point, a set of points, or a bounding box. SAM can effectively segment unseen object types in a variety on unseen contexts. Although CLIP has been previously employed for zero-shot point cloud understanding [42,58], where visual features are transferred to 3D points for downstream tasks, to the best of our knowledge, there are no existing object 6D pose estimation methods that lift 2D visual features to 3D point clouds. In this work, we successfully achieve this by using DINOv2 visual features to boost geometric features for registration, and for effectively disambiguating object poses of geometrically symmetric objects.

Zero-shot object 6D pose estimation methods, while less numerous compared to fully supervised [18,30,51] or few-/one-shot learning approaches [5,15,49], are gaining attention. In the initial step, zero-shot image segmentation techniques like SAM [25] can be employed to locate the object within the image. Zephyr [38] generates candidate poses for an object and selects the best one based on point-level differences in color and geometric domains. MegaPose [26] estimates object poses by rendering multiple views of the CAD model and matching these with the masked image to obtain a coarse pose. ZeroPose [6] uses multi-resolution geometric features to match regions between the scene's point cloud and the CAD model. Nguyen et al. [37] train a network to compute local features and match the image against rendered object templates. Similarly, ZS6D [3] matches DINOv2 [39] visual features against a database of features from rendered object templates, followed by a RANSAC-based PnP algorithm for final pose estimation. SAM6D [31] generates mask proposals from images using SAM and ranks them based on a combined score of semantics, appearance, and geometry against rendered object templates. Top proposals are then matched using 3D-3D correspondence matching.

A coarse point matching stage uses sparse correspondences to estimate the initial pose, followed by a fine point matching stage that refines it using Sparse-to-Dense Point Transformers. GigaPose [35] renders templates to extract dense features using ViT and then finds the best template using fast nearest neighbour search in the feature space. On top, two lightweight MLPs estimate 2D scale and in-plane rotation from a single 2D-2D correspondence using local features. FoundPose [40] first leverages DINOv2 and bag-of-words descriptors to shortlist similar templates. 2D-3D correspondences are then established among query and the selected templates using DINOv2 patch level features. FoundationPose [52] offers a unified approach for pose tracking and estimation in both model-based and model-free scenarios. It trains on a large synthetic dataset generated via Large Language Models (LLMs). The process starts with coarse pose estimation, refined by a transformer based architecture. The refined pose hypotheses are then evaluated by a pose selection network using a hierarchical comparison and a pose ranking encoder trained on pose-conditioned triplet loss.

Unlike existing zero-shot 6D pose estimation approaches, FreeZe employs deep neural networks that have been trained for various and different generic downstream tasks. FreeZe can utilize point cloud descriptors that were trained for point cloud registration, such as GeDi [43], and image feature extraction networks that were self-supervised on large-scale internet data, such as DINOv2 [39]. FreeZe is designed to be adaptable, allowing for the incorporation of new state-of-the-art feature representation methods as they become available in the future, without requiring re-training on specific data, all while maintaining the core methodology.

FoundationPose leverages LLMs to generate training synthetic data at scale. We make a step forward, and by leveraging recent geometric and vision foundation models, we do not require any training data at all. ZS6D and FoundPose are training-free competitors. However, unlike them that use visual features to find 2D-2D and 2D-3D correspondences, respectively, we exploit the depth information of the scene and we work with 3D-3D correspondences. Moreover, we leverage also geometric and not only vision foundation models.

3 Our approach

3.1 Overview

Given the 3D model of a *query object* Q and an RGBD image $\mathbf{I} \in \mathbb{R}^{H \times W \times 4}$ capturing Q, our goal is to find the 6D pose of Q with respect to the camera's reference frame. Let the *target object* T be the instance of Q captured in $\mathbf{I}$, our goal is then to estimate the 6DoF transformation $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ that relates Q to T, where $\mathbf{R} \in \mathbb{R}^{3\times3}$ and $\mathbf{t} \in \mathbb{R}^3$ are the rotation and translation components. We address the problem by leveraging geometric and vision foundation models to compute features for Q and T, and by then estimating $\mathbf{R}$ and $\mathbf{t}$ from 3D-3D correspondences (Fig. 2). We sample N points from the 3D model's surface to generate a point cloud $\mathcal{P}^Q \in \mathbb{R}^{N\times3}$ from which we compute geometric features for each point through a frozen pre-trained geometric encoder Ψ. Concurrently, we use the 3D model to render templates from different viewpoints, which we feed

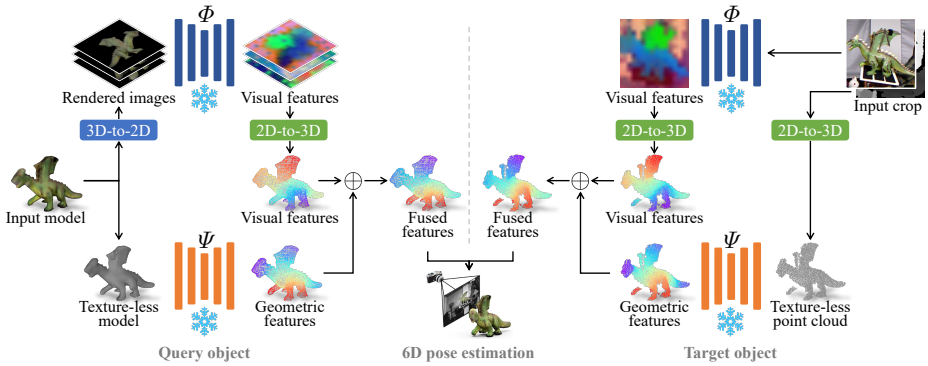

Fig. 2: Overview of FreeZe. Left-hand side, we create rendered images from the input model of the query object to extract visual features, which we then back project to the object point cloud. Concurrently, we extract geometric features, which we then fuse with the visual features. Similarly, right-hand side, we compute visual and geometric features of the target object imaged in the input crop, and fuse the two as before. Although the query and target objects are from two different modalities (a 3D model the former, and a RGBD image the latter), we employ the same vision and geometric encoders to compute their features. Lastly, we input the fused features to a registration algorithm based on feature matching to estimate the object 6D pose. The color heatmaps used for the features are generated by reducing feature dimensionality with t-SNE [34]. The symbol $\oplus$ indicates feature concatenation.

to a frozen pre-trained vision encoder Φ to compute pixel-level visual features that we subsequently back-project onto $\mathcal{P}^Q$ (Sec. 3.2). For the RGBD image, we locate T in $\mathbf{I}$ using segmentation masks estimated by a zero-shot object detectors. We generate a point cloud from the pixels belonging to the mask by using the depth channel and the camera's intrinsic parameters. We sample M points to get $\mathcal{P}^T \in \mathbb{R}^{M\times 3}$ and we compute geometric features for these points. Concurrently, we compute pixel-level visual features from the RGB channels and back-project them onto $\mathcal{P}^T$ (Sec. 3.3). The geometric and visual features, for both Q and T, are fused together (Sec. 3.4) and used to perform 3D-3D matching and registration (Sec. 3.5). Finally, we refine the pose of objects using our novel symmetry aware refinement algorithm (Sec. 3.6).

3.2 Query object processing

We compute the geometric features of Q by processing $\mathcal{P}^Q$ directly with the geometric encoder Ψ. Let $\mathbf{g}_n^Q \in \mathbb{R}^G$ be the feature of the n-th point of Q, where G is the feature size, then $\mathcal{G}^Q = \{\mathbf{g}_n^Q\}_{n=1}^N = \Psi(\mathcal{P}^Q)$ represents Q's geometric features. We compute the visual features of Q by rendering RGBD images of the textured 3D model of Q from R different viewpoints, to obtain the set $\{\mathbf{I}_r\}_{r=1}^R$. We also render the associated object segmentation masks, which we use to crop the portion $\mathbf{I}_r^Q$ of $\mathbf{I}_r$ occupied by Q. We process the RGB channels of $\mathbf{I}_r^Q$ with

the vision encoder Φ to generate pixel-level[3] visual features $\mathbf{V}_r^Q$. $\mathbf{V}_r^Q$ are then back-projected to the 3D model, and associated with $\mathcal{P}^Q$'s points as follows. First, we compute the correspondences between the RGB pixels of $\mathbf{I}_r^Q$ and the points of $\mathcal{P}^Q$. We convert the depth channel of $\mathbf{I}_r^Q$ into a viewpoint-dependent point cloud $\mathcal{P}_r^Q$ using the renderer's camera intrinsic parameters, and employ nearest neighbour search between the points of $\mathcal{P}_r^Q$ and $\mathcal{P}^Q$. Then, we leverage this correspondence map to associate the visual features $\mathbf{V}_r^Q$ extracted from the r-th rendered image $\mathbf{I}_r^Q$ to the points of $\mathcal{P}^Q$. Lastly, we aggregate multi-view features $\{\mathbf{V}_r^Q\}_{r=1}^R$ into a single set $\mathcal{V}^Q = \{\mathbf{v}_n^Q\}_{n=1}^N$ by averaging the contribution of each viewpoint as $\mathbf{v}_n^Q = \sum_{r=1}^R \mathbf{v}_{r,n}^Q / R$.

3.3 Target object processing

Localisation. We use a zero-shot object segmentation algorithm, such as CNOS [36], to localise T within $\mathbf{I}$. CNOS computes region proposals generated using SAM [25] (or FastSAM [56]) within $\mathbf{I}$, and compares them against object templates derived from the textured 3D model of query object. It assigns a score based on the visual feature similarity with the object templates to each proposal. CNOS produces a set of masks as output, each one with a respective confidence score. The common practice is to select the mask with the highest confidence score. However, we experimentally observed that CNOS is not well calibrated, as more accurate segmentation masks might have lower confidence scores. Hence, we keep the most confident masks as possible target object candidates. For each candidate, besides the mask we also extract the minimum bounding box $\mathbf{I}^T \in \mathbf{I}$ that contains the mask.

Feature extraction. We compute the geometric features of T by processing $\mathcal{P}^T$ with Ψ. Let $\mathbf{g}_m^T \in \mathbb{R}^G$ be the feature of the m-th point of T, and $\mathcal{G}^T = \{\mathbf{g}_m^T\}_{m=1}^M = \Psi(\mathcal{P}^T)$ be the set of features of T, where M is the number of points.s We compute pixel-level features $\mathbf{V}^T$ by processing the input crop $\mathbf{I}^T$ with Φ. However, associating $\mathbf{V}^T$ with the points of $\mathcal{P}^T$ is more straightforward than the process required for the query object. Leveraging the one-to-one correspondences between the pixels of the RGB crop and the depth, we can transfer pixel-level features $\mathbf{V}^T$ to the points of $\mathcal{P}^T$ forming the set $\mathcal{V}^T = \{\mathbf{v}_m^T\}_{m=1}^M$. Despite Φ processes both foreground and background pixels of $\mathbf{I}^T$, we only transfer foreground features to $\mathcal{P}^T$.

3.4 Visual and geometric feature fusion

We fuse visual and geometric features through concatenation for both Q and T. We apply L_2 normalization to $\mathcal{V}^Q$ $\left(\mathcal{V}^T\right)$ and $\mathcal{G}^Q$ $\left(\mathcal{G}^T\right)$ independently to account for potential differences in norms and to balance their contribution in the final

[3] Since we use a Transformer for Φ, we use bilinear interpolation to convert patch-level features to pixel-level ones. For example, DINOv2 [39] outputs a 16×16 grid of patch-level features.

feature. Specifically, the features we produce have the following form:

$$\mathcal{F}^Q = \{\mathbf{f}_n^Q\}_{n=1}^N = \left\{ \left[\frac{\mathbf{v}_n^Q}{\|\mathbf{v}_n^Q\|_2} \middle| \frac{\mathbf{g}_n^Q}{\|\mathbf{g}_n^Q\|_2} \right] \right\}_{n=1}^N,$$
$$\mathcal{F}^T = \{\mathbf{f}_m^T\}_{m=1}^M = \left\{ \left[\frac{\mathbf{v}_m^T}{\|\mathbf{v}_m^T\|_2} \middle| \frac{\mathbf{g}_m^T}{\|\mathbf{g}_m^T\|_2} \right] \right\}_{m=1}^M,$$

where $\mathbf{f}_n^Q, \mathbf{f}_m^T \in \mathbb{R}^{V+G}$,$[\cdot|\cdot]$ is the concatenation and $\|\cdot\|_2$ is the L_2 norm operator. We found this simple fusion strategy working well in practice. However, alternative fusion strategies can be employed, and we leave this for future developments.

3.5 Pose estimation

We employ an off-the-shelf registration algorithm based on 3D-3D feature matching, like RANSAC [8], to robustly estimate the coarse transformation $\mathbf{T}_c$ from the pair $((\mathcal{P}^Q, \mathcal{F}^Q), (\mathcal{P}^T, \mathcal{F}^T))$. RANSAC operates by sampling triplets of points from $\mathcal{P}^Q$, and searching for their corresponding points in $\mathcal{P}^T$ by performing a nearest neighbour search within the fused feature space. False matches are pruned, while true matches are utilised to compute the transformation $(\mathbf{R}, \mathbf{t})$, thereby registering $\mathcal{P}^Q$ to $\mathcal{P}^T$, *i.e.* $\mathcal{P}^Q\mathbf{R} + \mathbf{t} \approx \mathcal{P}^T$. The pair $(\mathbf{R}, \mathbf{t})$ represents the predicted 6D pose of Q. We estimate the pose for each possible candidate T, and retain only the one with associated the highest number of registration inliers.

3.6 Symmetry-aware refinement

We use the Iterative Closest Point (ICP) algorithm [7] to refine $\mathbf{T}_\mathrm{c}$ at point level and obtain a finer transformation $\mathbf{T}_\mathrm{f}$. Although ICP is known to be sensitive to local minima, we find it effective to refine several of our initial poses that are close to the correct solution. Then, we use visual features computed by Φ to solve pose ambiguities that arise from geometric symmetries. Although our fused features incorporate visual information, query objects with geometric symmetries can still lead to local-minimum registration solutions. We mitigate this problem by introducing a novel symmetry-aware refinement based on rendering and visual features matching. Our aim is to adjust the pose for those objects for which the estimate pose would be correct without considering the texture, *i.e.* when $\mathcal{P}^Q$ and $\mathcal{P}^T$ match, but the texture does not.

Symmetry estimation aims to identify geometric symmetries of Q's 3D model, regardless of its texture. We build a set of rotation matrices by uniformly sampling the rotation space. Our estimated symmetries $\{\mathbf{R}_s \in SO(3)\}^S$ are then the rotations that minimize the Chamfer distance [57] between $\mathbf{R}_s\mathcal{P}^Q$ and $\mathcal{P}^Q$.

Symmetry selection. We render the RGB images $\{\mathbf{I}_s\}^S$ depicting the 3D model of Q in the poses $\{\mathbf{T}_f \circ \mathbf{R}_s\}^S$. We use the bounding box defined in Sec. 3.3 to crop images $\{\mathbf{I}_s^T\}^S$. We then process $\{\mathbf{I}_s^T\}^S$ with Φ to produce visual features $\{\mathbf{V}_s\}^S$. The best candidate is selected by computing the cosine similarity between

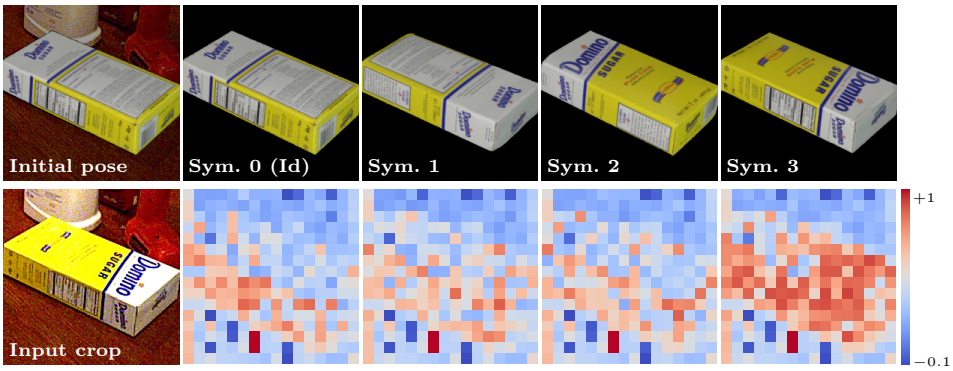

Fig. 3: Example of symmetry-aware refinement. We adjust the inaccurate initial pose (top left corner) by comparing the input crop (bottom left corner) with the rendered images of the four object symmetries (first row). Sym. 0 (Id) is the identity transformation, *i.e.* without applying any symmetric transformation. By computing the patch-level cosine similarities for each candidate (second row, red-white-blue colors), and averaging them to obtain a global score (third row), we can select the best candidate as the one with the highest score (✓).

$\{\mathbf{V}_s\}^S$ and the intermediate patch-level visual features $\mathbf{V}^T$ that we computed in Sec. 3.3. With Φ being a Transformer, we observed that patch-level features are more informative than the class token to encode pose information, a conclusion also supported in [36]. Moreover, as both visual features coming for I^T and I_s are at patch-level, we use them as they are. Instead, in Sec. 3.3, we needed to map features to points, hence we augmented patch-level features to pixel-level features. Fig. 3 shows an example of symmetry-aware refinement. Starting from an inaccurately estimated pose, we rotate and render the object according to its four estimated symmetries. We compute the patch-level cosine similarity between the visual feature of the input crop and the rendered images. Symmetry no. 3 exhibits a higher similarity score (+30%), enabling us to correct the initial pose.

4 Experiments

4.1 Experimental setup

We randomly sample $N = 5K$ points for the query object, and $M = 1K$ for the target object. We use the masks estimated with CNOS [36] and SAM6D [31] to localize the object within the input image. As geometric encoder we use GeDi [43], a geometric feature extractor for point clouds with strong generalization capability across different applications domains. To capture geometric details at different scales, we extract 32-dimensional GeDi features from local neighbours occupying 30% and 40% of the object's diameter, and concatenate them to obtain the final 64-dimensional geometric features. As vision encoder we use the ViT-giant

version of DINOv2 [39]. DINOv2 is a vision foundation model trained through self-supervision on large-scale web data. DINOv2 processes images of size 224×224 and outputs 1536-dimensional features. We apply PCA to reduce their dimensionality to $V = 64$, for efficiency and to match geometric features' dimensionality.

4.2 Datasets

We evaluate FreeZe on the seven core datasets of the BOP Benchmark [50]: LM-O [4], T-LESS [20], TUD-L [21], IC-BIN [11], ITODD [12], HB [24], and YCB-V [53]. Each dataset provides both 3D models of the objects and test RGBD images. **Object types.** T-LESS and ITODD contain industrial objects, such as electrical and mechanical items, with planar sides, sharp edges, and hollow parts. The other datasets contain ordinary objects, such as food packages, toy models, and stationery items, with finer geometric details and smooth boundaries. Industrial objects are provided as CAD models, while the 3D models of ordinary objects are reconstructed from images acquired with RGBD sensors. **Number of instances.** LM-O, TUD-L, HB, and YCB-V contain at most one instance per object, while the other datasets contain multiple instances. ITODD contains 4 instances per object on average, except a few images that contain 84 instances per object. T-LESS and IC-BIN contains 1.3 and 9 instances per object on average, respectively. **Photometric information.** T-LESS and ITODD objects are texture-less. LM-O, TUD-L, and HB objects have mostly uniform colors. IC-BIN and YCB-V objects have rich textures. **Noise levels.** TUD-L contains a single object for each scene with difficult light conditions. LM-O scenes are highly cluttered (other objects are present) and feature occlusions. IC-BIN scenes contain several instances of the same object, thus featuring different levels of occlusions and a high localization ambiguity. The other datasets contain mildly occluded objects.

4.3 Metrics

We evaluate the accuracy of pose estimation using the default metrics of the BOP Benchmark [50], which are VSD, MSSD, and MSPD. These metrics measure different errors between two 3D models obtained by transforming the object 3D model with the predicted and ground-truth poses. VSD measures the discrepancy between the depth maps obtained by rendering them. MSSD and MSPD measure the maximum Euclidean distance and the maximum reprojection error between their points, respectively. Both MSSD and MSPD take into consideration symmetries, by assuming the minimum value reached across all symmetrically equivalent ground-truth poses. For each dataset, we report the average of the three metrics, namely Average Recall (AR).

4.4 Quantitative results

Tab. 1 reports results and comparative analyses with previous and concurrent published works on the seven core datasets of the BOP Benchmark. All the

Table 1: Results on the BOP Benchmark datasets. We report the AR score on each of the seven core datasets of the BOP Benchmark and the mean AR across datasets. For a fair comparison, we report results with zero-shot localization prior, while we do not report any result with supervised or undefined prior. **Bold** font indicates best AR. Our results are highlighted. Keys. Training free: task-specific training free; Prior: type of instance localization prior; Refin.: pose refinement; Mean: Mean AR; '-': not available.

	Method	Training free	Input	Prior	Refin.	BOP Dataset LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	Mean
1	MegaPose [26]		RGB	CNOS		22.9	17.7	25.8	15.2	10.8	25.1	28.1	20.8
2	ZS6D [3]	✓	RGB			29.8	21.0	-	-	-	-	32.4	-
3	GigaPose [35]		RGB			29.9	27.3	30.2	23.1	18.8	34.8	29.0	27.6
4	FoundPose [40]	✓	RGB			39.7	33.8	46.9	23.9	20.4	50.8	45.2	37.3
5	SAM6D [31]		RGBD			57.0	38.2	69.8	41.5	41.4	66.9	73.2	55.4
6	FreeZe (ours)	✓	RGBD			**64.7**	**49.3**	**86.1**	**44.3**	**49.2**	**75.7**	**78.7**	**64.0**
7	MegaPose [26]		RGB	CNOS	✓	56.0	50.8	68.7	41.9	34.6	70.6	62.0	54.9
8	GigaPose [35]		RGB		✓	59.9	**57.0**	63.5	46.7	39.7	72.2	66.3	57.9
9	FoundPose [40]	✓	RGB		✓	61.0	**57.0**	69.3	47.9	40.7	72.3	69.0	59.6
10	ZeroPose [6]		RGBD		✓	53.8	40.0	83.5	39.2	52.1	65.3	65.3	57.0
11	MegaPose [26]		RGBD		✓	62.6	48.7	85.1	46.7	46.8	73.0	76.4	62.8
12	SAM6D [31]		RGBD		✓	63.5	46.3	80.0	46.5	54.3	71.1	80.0	63.2
13	FreeZe (ours)	✓	RGBD		✓	**69.0**	52.0	**93.6**	**49.9**	**56.1**	**79.0**	**85.3**	**69.3**
14	SAM6D [31]		RGBD	SAM6D		62.7	42.0	77.7	**50.4**	45.5	68.9	74.3	60.2
15	FreeZe (ours)	✓				**67.6**	**50.0**	**88.1**	48.7	**52.0**	**76.1**	**77.4**	**65.7**
16	SAM6D [31]		RGBD	SAM6D	✓	68.7	49.8	87.4	**56.1**	57.7	75.4	82.8	68.2
17	FreeZe (ours)	✓			✓	**71.6**	**53.1**	**94.9**	54.5	**58.6**	**79.6**	**84.0**	**70.9**

reported methods use a localization prior estimated in a zero-shot fashion, *i.e.* we only report methods that operate in the same setting for a fair comparison. The table is divided into two sections: the top part lists methods with CNOS localization priors, while the bottom part assumes SAM6D segmentation priors. FreeZe consistently outperforms all the works by a considerable margin. Specifically, when comparing with methods using CNOS masks and not using any pose refinement, FreeZe achieves +8.6% over the second-best method SAM6D [31] (row 5). When comparing against methods using CNOS masks and pose refinement, FreeZe outperforms the second-best method SAM6D (row 12) by +6.1%. We can observe a significant improvement in TUD-L, where FreeZe achieves a +8.5% with respect to MegaPose [26] (row 11) and a +13.6% with respect to SAM6D (row 12). TUD-L stands as a dataset renowned for its exceptionally challenging lighting conditions. In this context, FreeZe stands out by demonstrating the remarkable efficacy of leveraging geometric features to their fullest extent. Interestingly, FreeZe without any refinement is even better than the top competitor with refinement SAM6D (row 6 vs row 12). On average, our refinement step contributes to an increase of +5.2%. In the second part of the table, we compare FreeZe with SAM6D using their own estimated masks (SAM6D masks). Also in this setting, we establish state-of-the-art results for zero-shot object 6D pose estimation, both with and without any refinement. Finally, FreeZe using CNOS masks outperforms SAM6D with SAM6D masks by +1% (row 13 vs row 16), which is remarkable because SAM6D masks are more accurate than CNOS ones.

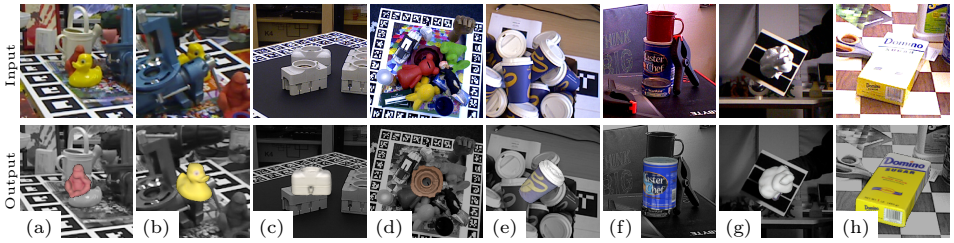

Fig. 4: Qualitative results of challenging cases. The top row shows input images, while the bottom row shows FreeZe's predictions by overlaying the object 3D model transformed according to the predicted pose. We use a grayscale version of the input image for a better contrast. Left to right: (a,b) highly-occluded texture-less toy models, (c,d) highly-occluded texture-less industrial items, (e,f) common objects with cylindrical shapes and informative textures, (g,h) difficult light conditions (strong shadows and a very bright capture). Despite these challenges, FreeZe can always find the correct pose.

4.5 Qualitative results

Fig. 4 shows examples of qualitative results, which include several challenges such as occlusions, clutter, variations in background and illumination conditions, poor object textures, severe object symmetries, and similarity with other objects or different instances of the same object. Fig. 4(a,b) show highly-occluded texture-less objects (the toy models of a red ape and a yellow duck); Fig. 4(c,d) show highly-occluded texture-less symmetric objects (a white electrical outlet and a bronze gear). To estimate the poses of (a,b,c,d), FreeZe relies mostly on geometric information, since these objects have non-informative uniform textures. Fig. 4(e,f) show objects with cylindrical shapes (a coffee cup and a food can), where we can observe the presence of multiple instances of the same coffee cup, each with different levels of occlusions. To estimate the poses of (e,f), FreeZe relies mostly on visual information, since cylindrical shapes present an infinite number of symmetries, and the correct pose can be established only by relying on texture information. Lastly, Fig. 4(g,h) show images captured under challenging light conditions. The uniformly-colored white toy frog in (g) is affected by shadows, while (h) shows a bright capture of a textured sugar box.

Fig. 5 shows example of failure cases. In Fig. 5(a), the red foam block is aligned only to the closest corner, with a pose nearly perpendicular to the correct one. This can be due to the part of the object that is occluded by the marker. In Fig. 5(b), the yellow tea box pose is predicted laying on its side rather than on its bottom. Because of the object's simple geometry and poor texture, we believe that aligning most of the corners of the box regardless of its texture can be acceptable for FreeZe. In Fig. 5(c), the main body of the red mug is fairly aligned, except the handle. This may be due to the limited surface area of the handle that causes FreeZe to prioritize aligning the rest of the object. In Fig. 5(d), the pose of the green drill handle is incorrectly estimated. The handle is fully occluded in the input image, so that predicting a correct pose is challenging without additional prior information. The visible portion (main body) is correctly aligned.

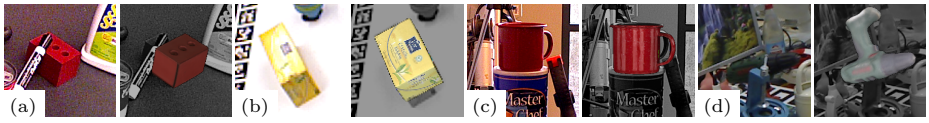

Fig. 5: Failure cases. FreeZe's predictions follow the relative input images. Left to right: (a) the foam block is aligned only to the closest corner, in a pose that is almost perpendicular to the correct one, (b) the tea box pose is predicted as if it was laying on its side rather than on its bottom, (c) the main body of the mug is nearly aligned, but the handle is not, (d) the pose of the drill handle (occluded) is incorrectly estimated.

4.6 Ablation study

FreeZe integrates geometric and visual features, making the choice of specific encoders both critical and carefully considered. As presented in Tab. 2, we conduct a thorough ablation study on LM-O [4] to ensure the optimal selection and to assess each component of FreeZe. Unless otherwise specified, we do not apply any pose refinement. Tab. 3 shows the ablation study on the refinement step. In both cases, results are reported in terms of AR. We use as localization prior CNOS segmentation masks across all the experiments.

Feature types. We assess the fusion of geometric and visual features. In rows 1–3 of Tab. 2, we evaluate different geometric feature extraction methods: hand-crafted approach FPFH [47], and learning-based techniques FCGF [9] and GeDi [43]. In rows 9–11 of Tab. 2, we evaluate different vision encoders: simple RGB values, and the foundation models CLIP [46] and DINOv2 [39]. We observed that: (i) GeDi is highly effective across scenes (+34.5 in AR w.r.t. FCGF); (ii) DINOv2 outperforms CLIP by about +6 in AR despite they are both based on ViT-L; (iii) geometric features outperforms visual ones by a significant margin.

Multi-scale geometric feature fusion. GeDi [43] uses a single radius to define the extent of local descriptors. Instead, we use multiple radii to encode information at multiple scales. In rows 4–8 of Tab. 2, we compare the original single-scale version of GeDi with the proposed multi-scale fusion of GeDi features for different radii. Radii are expressed as a ratio of the object diameter, *i.e.* 0.2 corresponds to 20% of the object diameter. The best AR is achieved when using multiple scales. Another crucial advantage of using multiple scales lies in its generalizability. We experimentally observed that the best-performing single-scale radius is dataset dependent, whereas using multiple scales allow us to use the same configuration consistently across all the seven core datasets.

Visual features backbone. In rows 12–15 of Tab. 2, we assess different DINOv2 ViT backbones: small (S), base (B), large (L), and giant (G) models. The latter achieves the highest AR (+2.8 in AR w.r.t. ViT-L, the CLIP backbone).

Localization prior type. In rows 16–19 of Tab. 2, we test a bounding box localization prior rather than a segmentation mask prior. Without using any pose refinement, bounding boxes and segmentation masks produce the same results. However, once when refine the coarse pose using ICP, FreeZe performs betters with the segmentation masks as localization prior instead of the bounding boxes.

Table 2: Ablation study on the LM-O [4] dataset. We individually asses different components of FreeZe. MS-GeDi is our proposed multi-scale GeDi. Keys. Prior: type of instance localization prior; m: segmenation mask; bb: bounding box; '-': not available.

		Geometric encoder		Vision encoder		Prior	ICP	AR
		Method	Radius	Method	Backbone			
Geometry	1	FPFH	0.3	-	-	m		20.7
	2	FCGF	-	-	-	m		20.8
	3	GeDi	0.3	-	-	m		**55.3**
	4	GeDi	0.2	-	-	m		52.6
	5	GeDi	0.3	-	-	m		55.3
	6	GeDi	0.4	-	-	m		55.0
	7	MS-GeDi	(0.2, 0.3)	-	-	m		56.2
	8	MS-GeDi	(0.3, 0.4)	-	-	m		**56.5**
Vision	9	-	-	RGB	-	m		15.1
	10	-	-	CLIP	ViT-L	m		30.5
	11	-	-	DINOv2	ViT-L	m		**36.8**
	12	-	-	DINOv2	ViT-S	m		31.9
	13	-	-	DINOv2	ViT-B	m		33.7
	14	-	-	DINOv2	ViT-L	m		36.8
	15	-	-	DINOv2	ViT-G	m		**39.6**
Prior	16	MS-GeDi	(0.3, 0.4)	DINOv2	ViT-G	bb		64.7
	17	MS-GeDi	(0.3, 0.4)	DINOv2	ViT-G	bb	✓	68.6
	18	MS-GeDi	(0.3, 0.4)	DINOv2	ViT-G	m		64.7
	19	MS-GeDi	(0.3, 0.4)	DINOv2	ViT-G	m	✓	**69.0**

Table 3: Ablation study on the pose refinement module. We report the AR scores achieved without any refinement, with only an ICP-based refinement and with the complete ICP+SAR refinement on the seven core datasets of the BOP Benchmark. Keys. Refin.: pose refinement; Mean: Mean AR.

	Refin.	LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	Mean
1	None	64.7	49.3	86.1	44.3	49.2	75.7	78.7	64.0
2	+ICP	**69.0**	**52.0**	**93.6**	47.3	**56.1**	78.4	84.9	68.8
3	+SAR	**69.0**	**52.0**	**93.6**	**49.9**	**56.1**	**79.0**	**85.3**	**69.3**

This suggests that our coarse pose estimation is robust to background variations, while the ICP-based refinement step is sensitive to background outliers.

Pose refinement. In Tab. 3, we evaluate the effectiveness of the pose refinement by comparing the coarse estimation (row 1) with the pose obtained using an ICP-based refinement (row 2), and the one obtained adding also our Symmetry-Aware Refinement (SAR for short, row 3). SAR gains the most improvement over the basic ICP on datasets containing geometrically symmetric objects with discriminant texture, particularly on IC-BIN (+2.6 in AR), HB (+0.6 in AR) and YCB-V(+0.4 in AR). Instead, on datasets like LM-O, T-LESS, TUD-L and ITODD, SAR does not provide any advantage compared to solely relying on ICP.

5 Conclusions

We presented FreeZe, a novel approach to zero-shot object 6D pose estimation that leverages the strengths of pre-trained geometric and vision foundation models without the need for training on task-specific data. By leveraging 3D geometric features and 2D visual features to create discriminative 3D point-level descriptors, FreeZe outperforms all competitors on the BOP Benchmark, which includes seven datasets, setting a new state-of-the-art bar for the field. However, FreeZe faces certain **limitations**, which include the large size of foundation models that may restrict their deployment on edge devices, limiting real-world applicability in certain scenarios. One can mitigate this by using distillation techniques. Another **future research** direction includes improving our 3D registration process, possibly by advancing beyond simple RANSAC algorithms.

Acknowledgements

This work was supported by the European Union's Horizon Europe research and innovation programme under grant agreement No 101058589 (AI-PRISM), and by the PNRR project FAIR – Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

References

1. 6d localization of unseen objects – core datasets. https://bop.felk.cvut.cz/leaderboards/pose-estimation-unseen-bop23/core-datasets/ (2024), accessed: 1st March 2024
2. Ao, S., Hu, Q., Yang, B., Markham, A., Guo, Y.: Spinnet: Learning a general surface descriptor for 3d point cloud registration. In: CVPR (2021)
3. Ausserlechner, P., Haberger, D., Thalhammer, S., Weibel, J.B., Vincze, M.: Zs6d: Zero-shot 6d object pose estimation using vision transformers. arXiv:2309.11986 (2023)
4. Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6D object pose estimation using 3D object coordinates. In: ECCV (2014)
5. Castro, P., Kim, T.K.: Posematcher: One-shot 6d object pose estimation by deep feature matching. arXiv:2304.01382 (2023)
6. Chen, J., Sun, M., Bao, T., Zhao, R., Wu, L., He, Z.: ZeroPose: CAD-model-based zero-shot pose estimation. arXiv:2305.17934 (2023)
7. Chen, Y., Medioni, G.: Object modelling by registration of multiple range images. Image and vision computing (1992)
8. Choi, S., Zhou, Q.Y., Koltun, V.: Robust reconstruction of indoor scenes. In: CVPR (2015)
9. Choy, C., Park, J., Koltun, V.: Fully convolutional geometric features. In: ICCV (2019)
10. Corsetti, J., Boscaini, D., Poiesi, F.: Revisiting Fully Convolutional Geometric Features for object 6D pose estimation. In: ICCV-W (2023)
11. Doumanoglou, A., Kouskouridas, R., Malassiotis, S., Kim, T.K.: Recovering 6D object pose and predicting next-best-view in the crowd. In: CVPR (2016)
12. Drost, B., Ulrich, M., Bergmann, P., Hartinger, P., Steger, C.: Introducing MVTec ITODD – A dataset for 3D object recognition in industry. In: ICCV-W (2017)
13. Gao, N., Ngo, V.A., Ziesche, H., Neumann, G.: Sa6d: Self-adaptive few-shot 6d pose estimator for novel and occluded objects. In: CoRL (2023)
14. Girdhar, R., El-Nouby, A., Liu, Z., Singh, M., Alwala, K.V., Joulin, A., Misra, I.: Imagebind: One embedding space to bind them all. In: CVPR (2023)
15. Goodwin, W., Havoutis, I., Posner, I.: You only look at one: Category-level object representations for pose estimation from a single example. In: CoRL (2023)
16. Goodwin, W., Vaze, S., Havoutis, I., Posner, I.: Zero-shot category-level object pose estimation. In: ECCV (2022)
17. He, X., Sun, J., Wang, Y., Huang, D., Bao, H., Zhou, X.: Onepose++: Keypoint-free one-shot object pose estimation without cad models. NeurIPS (2022)
18. He, Y., Huang, H., Fan, H., Chen, Q., Sun, J.: Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In: CVPR (2021)
19. He, Y., Wang, Y., Fan, H., Sun, J., Chen, Q.: Fs6d: Few-shot 6d pose estimation of novel objects. In: CVPR (2022)
20. Hodan, T., Haluza, P., Obdrzalek, S., Matas, J., Lourakis, M., Zabulis, X.: T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. In: WACV (2017)
21. Hodan, T., Michel, F., Brachmann, E., Kehl, W., Buch, A.G., Kraft, D., Drost, B., Vial, J., Ihrke, S., Zabulis, X., Sahin, C., Manhardt, F., Tombari, F., Kim, T.K., Matas, J., Rother, C.: BOP: Benchmark for 6D object pose estimation. In: ECCV (2018)
22. Hoque, S., Xu, S., Maiti, A., Wei, Y., Arafat, M.Y.: Deep learning for 6d pose estimation of objects—a case study for autonomous driving. Expert Systems with Applications (2023)

23. Huang, S., Gojcic, Z., Usvyatsov, M., Andreas Wieser, K.S.: Predator: Registration of 3d point clouds with low overlap. In: CVPR (2021)
24. Kaskman, R., Zakharov, S., Shugurov, I., Ilic, S.: HomebrewedDB: RGB-D dataset for 6D pose estimation of 3D objects. In: ICCV-W (2019)
25. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. arXiv:2304.02643 (2023)
26. Labbé, Y., Manuelli, L., Mousavian, A., Tyree, S., Birchfield, S., Tremblay, J., Carpentier, J., Aubry, M., Fox, D., Sivic, J.: Megapose: 6d pose estimation of novel objects via render & compare. arXiv:2212.06870 (2022)
27. Li, G., Zhu, D., Zhang, G., Shi, W., Zhang, T., Zhang, X., Li, J.: Sd-pose: Structural discrepancy aware category-level 6d object pose estimation. In: WACV (2023)
28. Li, L., Fu, H., Ovsjanikov, M.: Wsdesc: Weakly supervised 3d local descriptor learning for point cloud registration. IEEE Transactions on Visualization and Computer Graphics (2022)
29. Li, L., Zhu, S., Fu, H., Tan, P., Tai, C.L.: End-to-end learning local multi-view descriptors for 3D point clouds. In: CVPR (2020)
30. Li, Z., Wang, G., Ji, X.: Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In: ICCV (2019)
31. Lin, J., Liu, L., Lu, D., Jia, K.: SAM-6D: Segment anything model meets zero-shot 6D object pose estimation. arXiv preprint arXiv:2311.15707 (2023)
32. Lin, J., Wei, Z., Li, Z., Xu, S., Jia, K., Li, Y.: Dualposenet: Category-level 6d object pose and size estimation using dual pose network with refined learning of pose consistency. In: ICCV (2021)
33. Liu, J., Sun, W., Liu, C., Zhang, X., Fu, Q.: Robotic continuous grasping system by shape transformer-guided multi-object category-level 6d pose estimation. IEEE Transactions on Industrial Informatics (2023)
34. van der Maaten, L., Hinton., G.: Visualizing high-dimensional data using t-sne. Journal of Machine Learning Research (2008)
35. Nguyen, V.N., Groueix, T., Salzmann, M., Lepetit, V.: GigaPose: Fast and robust novel object pose estimation via one correspondence. arXiv preprint arXiv:2311.14155 (2023)
36. Nguyen, V.N., Hodan, T., Ponimatkin, G., Groueix, T., Lepetit, V.: CNOS: A strong baseline for CAD-based novel object segmentation. In: ICCV-W (2023)
37. Nguyen, V.N., Hu, Y., Xiao, Y., Salzmann, M., Lepetit, V.: Templates for 3D Object Pose Estimation Revisited: Generalization to New Objects and Robustness to Occlusions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
38. Okorn, B., Gu, Q., Hebert, M., Held, D.: Zephyr: Zero-shot pose hypothesis rating. In: ICRA (2021)
39. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv:2304.07193 (2023)
40. Örnek, E.P., Labbé, Y., Tekin, B., Ma, L., Keskin, C., Forster, C., Hodan, T.: FoundPose: Unseen Object Pose Estimation with Foundation Features. arXiv preprint arXiv:2311.18809 (2023)
41. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: Pvnet: Pixel-wise voting network for 6dof pose estimation. In: CVPR (2019)
42. Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T., et al.: Openscene: 3d scene understanding with open vocabularies. In: CVPR (2023)

43. Poiesi, F., Boscaini, D.: Learning general and distinctive 3D local deep descriptors for point cloud registration. IEEE TPAMI (2023)
44. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017)
45. Rad, M., Oberweger, M., Lepetit, V.: Feature mapping for learning fast and accurate 3d pose inference from synthetic images. In: CVPR (2018)
46. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning (2021)
47. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: ICRA (2009)
48. Su, Y., Rambach, J., Minaskan, N., Lesur, P., Pagani, A., Stricker, D.: Deep multi-state object pose estimation for augmented reality assembly. In: ISMAR-Adjunct (2019)
49. Sun, J., Wang, Z., Zhang, S., He, X., Zhao, H., Zhang, G., Zhou, X.: Onepose: One-shot object pose estimation without cad models. In: CVPR (2022)
50. Sundermeyer, M., Hodaň, T., Labbe, Y., Wang, G., Brachmann, E., Drost, B., Rother, C., Matas, J.: Bop challenge 2022 on detection, segmentation and pose estimation of specific rigid objects. In: CVPR (2023)
51. Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S.: Densefusion: 6d object pose estimation by iterative dense fusion. In: CVPR (2019)
52. Wen, B., Yang, W., Kautz, J., Birchfield, S.: FoundationPose: Unified 6D pose estimation and tracking of novel objects. arXiv preprint arXiv:2312.08344 (2023)
53. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: PoseCNN: A Convolutional Neural Network for 6D object pose estimation in cluttered scenes. In: Robotic Science and Systems (2018)
54. Yang, J., Zhang, Q., Xiao, Y., Cao, Z.G.: TOLDI: An effective and robust approach for 3D local shape description. Patt. Recogn. (2017)
55. Zhao, H., Zhuang, H., Wang, C., Yang, M.: G3doa: Generalizable 3d descriptor with overlap attention for point cloud registration. RA-L (2022)
56. Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M., Tang, M., Wang, J.: Fast segment anything. arXiv:2306.12156 (2023)
57. Zhao, Y., Birdal, T., Deng, H., Tombari, F.: 3D point capsule networks. In: CVPR (2019)
58. Zhu, X., Zhang, R., He, B., Guo, Z., Zeng, Z., Qin, Z., Zhang, S., Gao, P.: Point-CLIPv2: Prompting CLIP and GPT for Powerful 3D Open-world Learning. In: ICCV (2023)
59. Zhuang, C., Li, S., Ding, H.: Instance segmentation based 6d pose estimation of industrial objects using point clouds for robotic bin-picking. Robotics and Computer-Integrated Manufacturing (2023)

Supplementary material for FreeZe: Training-free zero-shot 6D pose estimation with geometric and vision foundation models

Andrea Caraffa[1], Davide Boscaini[1], Amir Hamza[1,2], and Fabio Poiesi[1]

[1] Fondazione Bruno Kessler, Trento, Italy
{acaraffa,dboscaini,ahamza,poiesi}fbk.eu
[2] University of Trento, Italy

1 Introduction

We present the supplementary material in support of our main paper. The content is organized as follows:
- In Sec. 2, we evaluate the effectiveness of entangling geometric and visual features compared to using the two components independently.
- In Sec. 3, we highlight the role of the proposed Symmetry-Aware Refinement (SAR) in resolving ambiguous poses of geometrically symmetric objects.
- In Sec. 4, we investigate how object occlusions affect pose estimation accuracy.
- In Sec. 5, we provide an analysis of the computational time.
- In Sec. 6, we provide additional qualitative results on the seven core datasets of the BOP Benchmark [50].

2 Effectiveness of geometric and visual feature entanglement

We present qualitative results to illustrate the contribution of geometric and visual features within FreeZe. Fig. 1 showcases examples of poses predicted using exclusively geometric features (second row), exclusively visual features (third row), and their fusion (fourth row). In column (a), geometric features alone fail to accurately estimate the pose of the red cracker box, while visual features succeed in doing so. Conversely, in column (b), visual features alone struggle, while geometric features accurately predict the pose of the yellow banana. Finally, in column (c), both geometric and visual features fail to correctly estimate the pose of the red bowl when used individually. Through feature fusion, FreeZe capitalizes on the strengths of both feature types, enabling precise pose predictions for both (a) and (b). Remarkably, FreeZe achieves accurate pose estimation even when both feature types independently falter, as illustrated in column (c).

Our experimental observations indicate that geometric features extracted from objects with geometric symmetries, such as boxes and cylinders, can lead to two types of pose estimation errors. In the first case, the predicted pose might be inaccurate, as depicted in column (a). In the second case, although the object is

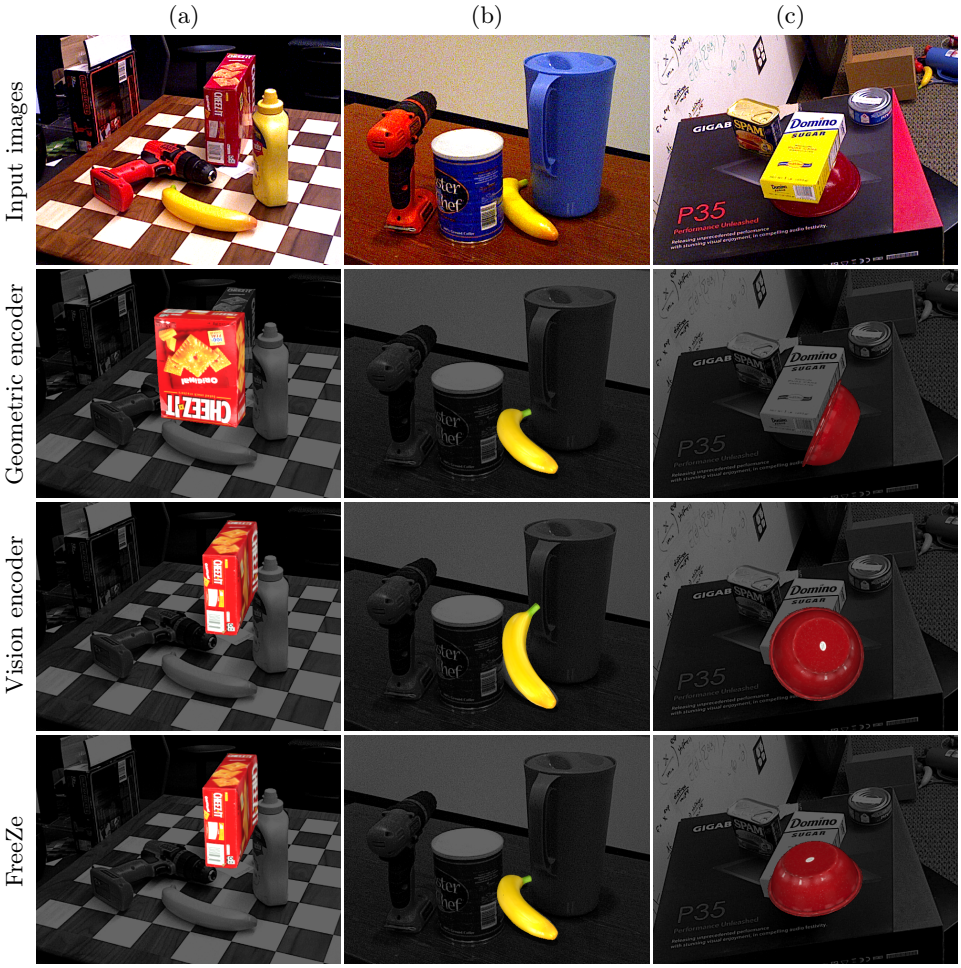

Fig. 1: Qualitative results illustrating the effectiveness of geometric and visual feature entanglement in FreeZe. Columns show three images from YCB-V [53]. Rows show, from top to bottom, the input image, the pose predicted using exclusively geometric features, the pose predicted using exclusively visual features, and the pose predicted using our fused features. Backgrounds are converted to grayscale to enhance contrast.

positioned correctly, it may have an incorrect rotation about one of its symmetry axes. The proposed Symmetry-Aware Refinement (SAR) module can only address errors of the second type but fails to solve errors of the first type. Therefore, the fusion of visual features with geometric ones is crucial for facilitating accurate pose predictions.

Fig. 2 visualizes geometric, visual, and our fused features in the RGB space by reducing their dimensionality via Principal Component Analysis. Interestingly, different instances of cans and boxes have similar geometric features, while this is not true for our fused features thanks to the integration with visual features.

Fig. 2: Visualization of the different types of features considered by FreeZe. Rows show different objects from YCB-V [53]. Columns show, from left to right, the 3D model of the object, the geometric features, the visual features, and our fused features.

3 Impact of Symmetry-Aware Refinement

We evaluate the effectiveness of our proposed Symmetry-Aware Refinement (SAR) module in correcting the ambiguous poses of geometrically symmetric objects. In Fig. 3, we illustrate the pose correction for various objects from YCB-V [53]. The first three rows depict cans of different types and levels of occlusion. SAR is capable of correcting minor errors in the estimated poses. The last two rows present cases where poses are incorrectly flipped by approximately 180 degrees. Such errors are infrequent, as the majority of poses are accurately estimated during the initial coarse pose estimation phase. In such instances, SAR successfully corrects the poses, despite significant rotational errors. Most of the poses of geometrically symmetric objects are correctly predicted thanks to the fusion with visual features, while about 15% of them are corrected using SAR.

4 Impact of object occlusions

The BOP Benchmark [50] includes challenging occlusion scenarios. We assess how this aspect affects FreeZe's performance. In Fig. 4(left), we compute the AR on LM-O [4], T-LESS [20], TUD-L [21], IC-BIN [11], and YCB-V [53] considering only target objects with visible surface greater than a given threshold. More precisely, we define the visible surface threshold as the minimum value below which we no longer consider the target in the computation of the AR. In Fig. 4(right) we also show the number of valid targets as percentage over the whole dataset for different visible surface thresholds. Given a target object in an image, we compute its visible surface as the Intersection over Union (IoU) of its ground-truth mask and visibility mask. A visible surface equal to 100% means the target is totally visible, while a visible surface equal to 0% means the target it totally occluded. As the minimum visible surface increases so does the AR, however, the performances on the datasets exhibit different trends. In TUD-L, occlusions are less relevant, i.e. nearly all targets have a visible surface larger than 80%. LM-O and IC-BIN are the most challenging datasets due to occlusions; only about 60% and 40% of the targets, respectively, have a visible surface larger than 80% (see Fig. 4(right)). Across all the datasets FreeZe has lower AR on IC-BIN and T-LESS. We can deduce that the low AR on IC-BIN is due its severe occlusions. FreeZe's performance on T-LESS, instead, is less related to occlusions since it presents additional challenges as the objects lack informative textures.

5 Analysis of computational time

In Tab. 1, we analyze the average execution time of FreeZe on TUD-L [21]. Our experiments are conducted using a NVIDIA Tesla A40 GPU and an Intel(R) Xeon(R) Silver 4316 CPU operating at 2.30GHz, utilizing 16 cores. We also reproduced the experiments of MegaPose [26][3] and SAM6D [31][4] on our hardware

[3] github.com/agimus-project/happypose, accessed Mar. 2024.
[4] github.com/JiehongLin/SAM-6D, accessed Mar. 2024

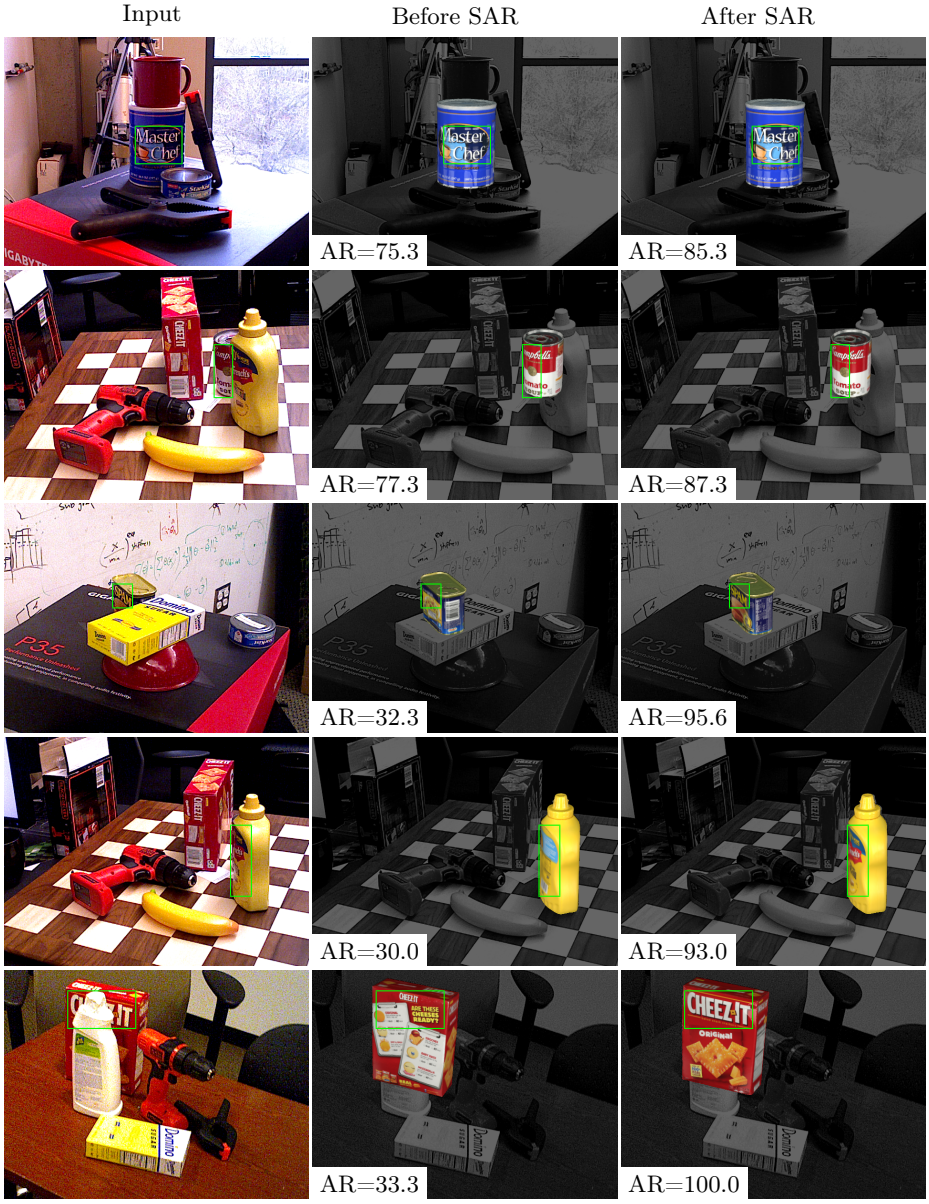

Fig. 3: Qualitative results for the proposed Symmetry-Aware Refinement (SAR) module on five images from YCB-V [53]. Rows show different examples. Columns show, from left to right, the input data, the coarse pose prediction, and the SAR-refined pose prediction. Backgrounds are converted to grayscale to enhance contrast. The overlaid green boxes highlight regions where pose correction by SAR is more visible.

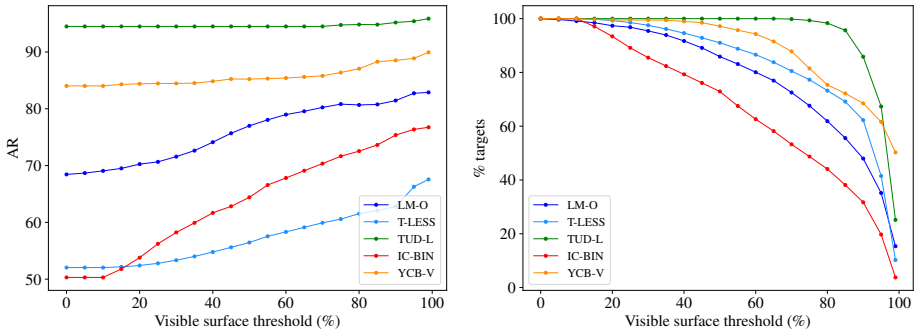

Fig. 4: Left: Average Recall (AR) for target objects selected according to different visible surface thresholds. The visible surface is computed as the IoU of the ground-truth mask and ground-truth visibility mask. The AR is reported on five core datasets of the BOP Benchmark. Right: Percentage of target objects as a function of different visible surface thresholds. The visible surface is computed as the IoU between the ground-truth mask and the ground-truth visibility mask. The targets percentage is reported on five core datasets of the BOP Benchmark.

Table 1: Comparison of computational times on TUD-L [21]. We compared FreeZe against MegaPose [26] and the concurrent work SAM6D [31]. To ensure a fair comparison, we reproduced all experiments on the same hardware.

Method	Input data		Localization prior		AR	Time (s)
	RGB	D	CNOS	SAM6D		
MegaPose [26]	✓		✓		65.9	4.59
MegaPose [26]	✓			✓	68.4	6.06
SAM6D [31]	✓	✓	✓		82.0	0.98
SAM6D [31]	✓	✓		✓	90.3	3.94
FreeZe (ours)	✓	✓	✓		93.6	2.72
FreeZe (ours)	✓	✓		✓	94.9	4.79

to ensure a fair comparison of the computational times. For a given 3D model and an image, FreeZe estimates the pose in 2.72 seconds using CNOS [36] masks. When using SAM6D masks, FreeZe takes 4.79 seconds. In both cases the timing for predicting the masks are included. Compared to MegaPose, using the same CNOS mask priors, FreeZe achieves a significant speedup of 41% and a gain of +27.7 in terms of Average Recall (AR). When using SAM6D masks, FreeZe still achieves a speedup of about 21% and a gain of +26.5 in AR. It is important to note that the tested version of MegaPose relies solely on RGB images, while FreeZe also incorporates depth information, surpassing MegaPose in both accuracy and speed. FreeZe achieves significantly higher accuracy (+4.6 AR) than SAM6D when utilizing SAM6D masks, despite being 0.85 seconds slower. When employing CNOS masks, FreeZe surpasses SAM6D by +11.6 AR, albeit being 1.78 seconds slower. Finally, FreeZe with CNOS masks surpasses SAM6D with its own masks, gaining a +3.3 AR, and being 1.22 seconds faster.

6 Additional qualitative results

We present qualitative results for each of the seven core datasets of the BOP Benchmark [50]. Each figure has the same structure. Rows show the methods: MegaPose [26] (second), SAM6D [31] (third) and FreeZe (fourth). Columns show different examples. To aid readers in analyzing the results, incorrect predictions are highlighted with **red** arrows, and missing predictions are emphasized using yellow arrows. Additionally, for better contrast against the RGB-colored objects, the backgrounds in the second to fifth rows are converted to grayscale. Across the seven datasets we highlight cases where FreeZe performs equivalently, surpasses, or falls short compared to the other methods. All the selected methods use masks estimated by CNOS [36].

LM-O dataset. Fig. 5 shows qualitative results on LM-O [4]. In column (a), MegaPose predictions for the white-black glue flip the pose by 180 degrees. This discrepancy is noticeable either by examining the shape portion highlighted with the red arrow, or by observing the mismatched side texture compared to the input image. The challenging aspect of this object lies in its near-symmetry: it has minimal differences in shape, and a subtle brand presence in the front texture, absent in the back texture. SAM6D correctly estimates the glue's pose, however, it sharply fails to predict the red primate's pose. In column (b), MegaPose prediction for the white watering can is quite inaccurate (red arrow). SAM6D does not predict any pose for the same object (yellow arrow). This scene is challenging due to heavy occlusion of the watering can's sprinkler. Additionally, SAM6D fails to predict the correct pose for the blue hole-puncher. In column (c), the white watering can's pose is incorrectly predicted by both MegaPose (less severe error) and SAM6D (upside-down), most likely because its handle is occluded. The pose for the blue hole-puncher is incorrectly predicted by MegaPose, presenting the object upside-down (evidenced by the red arrow tip pointing towards the top of the hole-puncher), despite the object being un-occluded. FreeZe consistently identifies the correct pose for all objects in all cases.

T-LESS dataset. Fig. 6 shows qualitative results on T-LESS [20]. In column (a), no methods can correctly predict all the object poses, with FreeZe only missing a single object pose (yellow arrow). In column (b), all methods but FreeZe miss one of the objects (yellow arrows), possibly because the instance localization priors provided by CNOS are too wide and include both background and other objects. In column (c), all methods except FreeZe inaccurately predict the pose of the largest item. SAM6D predicts the orientation of the electrical item as if it was lying on its side instead of its bottom, whereas MegaPose's prediction of the translation component notably deviates (the scale of the object is significantly different from the correct one). These inaccuracies likely result from the low quality of the instance localization prior, which in this case covers only a tiny portion of the object. Despite using the same localization priors, FreeZe predicts the correct pose for all objects in (b,c).

TUD-L dataset. Fig. 7 shows qualitative results on TUD-L [21]. It contains three different objects: a toy model of a green-brown dragon, a toy model of a white frog, and a white watering can. In column (a), FreeZe estimates are

accurate, while both MegaPose and SAM6D struggle to accurately estimate the dragon's pose. This issue arises because the CNOS localization prior is too wide, including also the person holding the object. This can be evinced by noticing that SAM6D's prediction is close to the person's leg (red arrow) rather than their hand. In column (b), FreeZe outperforms MegaPose, despite the associated localization prior only covering the frog's head. In column (c), FreeZe performs slightly worse than MegaPose. Our predicted pose leads to a moderate misalignment of the watering can's spout. This discrepancy occurs because the localization prior lacks coverage of the watering can's spout portion.

IC-BIN dataset. Fig. 8 shows qualitative results on IC-BIN [11]. It contains multiple instances of two object categories, a box-shaped juice carton and a cylindrical-shaped coffee cup, placed inside a bin. In column (a), we present a challenging scenario involving heavily-occluded coffee cups. MegaPose and SAM6D incorrectly predict the poses of the most visible objects and miss the occluded ones. FreeZe accurately predict the poses of all objects except missing the most occluded one (highlighted by the yellow arrow). Regarding one of the cups (indicated by the red arrow), FreeZe correctly aligns its cylindrical shape but faces challenges with aligning its texture (gold-colored brand logo). In column (b), we present a scenario involving five instances of the same juice carton. SAM6D makes one incorrect prediction while MegaPose makes two errors and misses one object. FreeZe outperforms all competitors by accurately estimating the pose of all five objects. In column (c), we consider both object types simultaneously. MegaPose and SAM6D perform poorly, with three and five errors, respectively, and multiple missing predictions. FreeZe yields poses with three errors and one missing prediction. Interestingly, for the juice carton in the middle, FreeZe correctly aligns the portion included in the instance localization prior, which consists only of the circular brand logo.

ITODD dataset. Fig. 9 shows qualitative results on ITODD [12]. It features 28 industrial objects captured from multiple views using a grayscale camera. The 3D models of texture-less objects are colored in light-green for a better contrast. In column (a), FreeZe predicts the correct pose for all the objects despite the presence of clutter, while MegaPose and SAM6D struggle on one of them (the one indicated by the red arrow). In column (b), FreeZe outperforms all the other methods by predicting the correct pose for all the objects. Instead, SAM6D has the lowest performance, correctly estimating only the pose of the most visible object. In addition to this object, MegaPose also correctly predict the pose of one of the other two less visible objects, even though the pose estimated by MegaPose deviates a bit from the ground-truth one. In column (c), we present a scene containing three instances of the same object without any occlusions. MegaPose and SAM6D fail to predict the pose of the top-right object, while FreeZe struggles with predicting the pose of bottom one. The errors of MegaPose, SAM6D and FreeZe are similar, as they rotate the object by 90 degrees. The estimated pose aligns with the two larger extremities, correctly matching that portion of the surface, while missing alignment with the smaller third extremity.

HB dataset. Fig. 10 shows qualitative results on HB [24]. It contains toys, industrial, and household items arranged in highly cluttered scenes. In column (a), MegaPose performs poorly, by wrongly estimating the poses of three out of seven objects, i.e. the black-white cow, the green rabbit, and the white car. SAM6D correctly estimates the poses of five objects but misses two (indicated by yellow arrows). FreeZe perform relatively good by correctly predicting the poses of six objects and missing one (the white car). In column (b), all competitors exhibit poor performance, each making three incorrect predictions. In contrast, FreeZe outperforms them by making only a single inaccurate prediction for the black box object (red arrow). In column (c), all other methods struggle with the blue "Jaffa cakes" box: SAM6D successfully locates the object but flips its pose by 180 degrees, whereas both MegaPose and FreeZe fail to even localize it.

YCB-V dataset. Fig. 11 shows qualitative results on YCB-V [53]. Columns show scenarios where FreeZe performs equivalently (a), surpasses (b), or falls short (c) compared to the other methods. In column (a), MegaPose and SAM6D incorrectly predict the pose of the heavily-occluded scissors, whereas FreeZe successfully determine the correct pose for every object. In column (b), MegaPose encounter difficulties in determining the pose of the brown box, heavily occluded by the red box. Conversely, SAM6D and FreeZe predict the correct pose for all objects despite the occlusions. In column (c), MegaPose predicts the wrong pose for the red bowl, despite the object being un-occluded. Both SAM6D and FreeZe struggle with the potted meat can, flipping the pose by 180 degrees. This discrepancy is evident in the mismatched side texture compared to the input image (highlighted by the red arrow, where the front texture lacks the "Spam" logo). However, this case is particularly interesting because of the object's 3D model inconsistency with the real product: the opening tab is positioned in a different location. We believe that both SAM6D and FreeZe prioritize aligning this visible geometric detail over the heavily-occluded side texture.

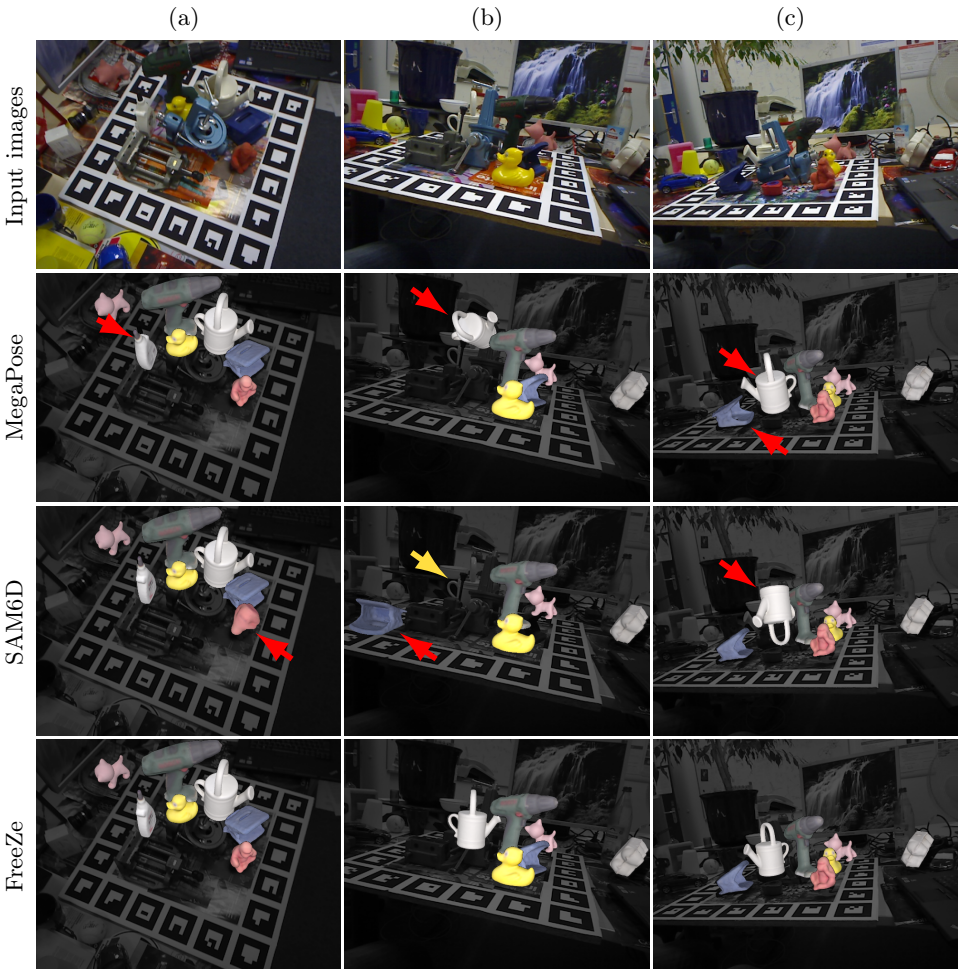

Fig. 5: Qualitative results on LM-O [4]. Columns show different examples. Rows show a comparison against different methods. **Red** (**yellow**) arrows highlight wrong (missing) predictions. Backgrounds are converted to grayscale for a better contrast.

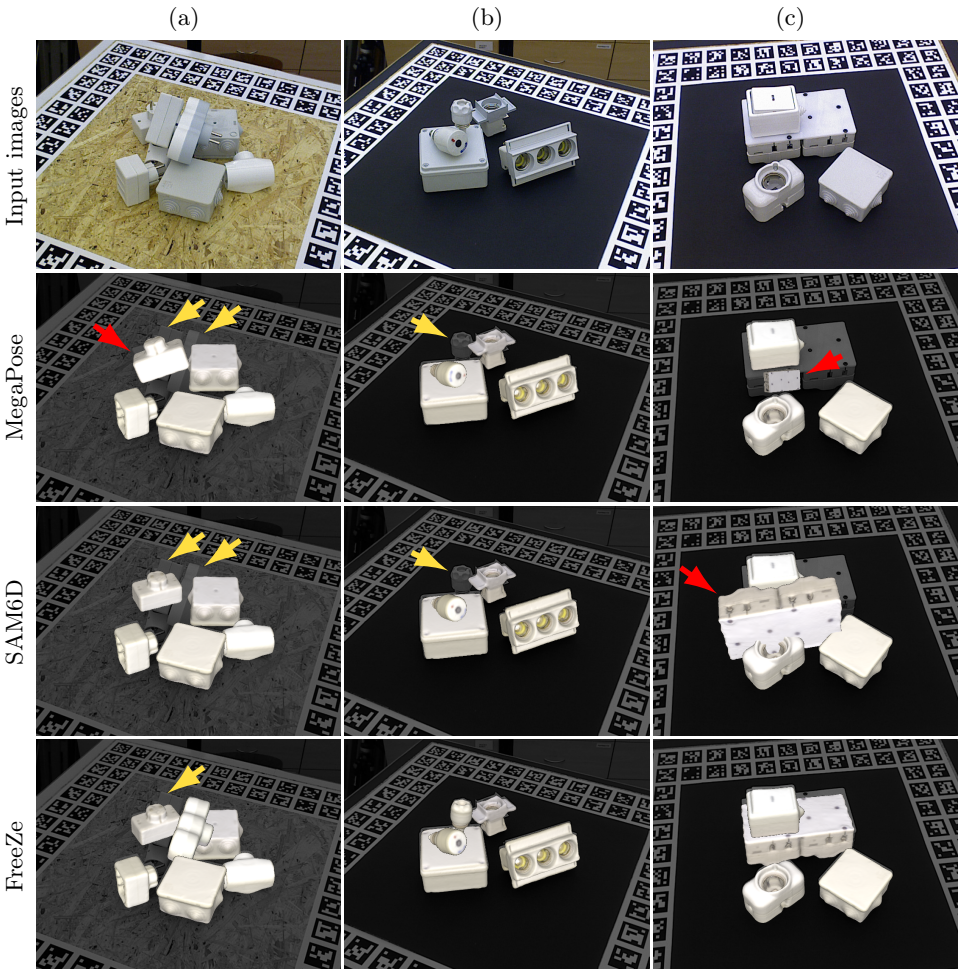

Fig. 6: Qualitative results on T-LESS [20]. Columns show different examples. Rows show a comparison against different methods. **Red** (yellow) arrows highlight wrong (missing) predictions. Backgrounds are converted to grayscale for a better contrast.

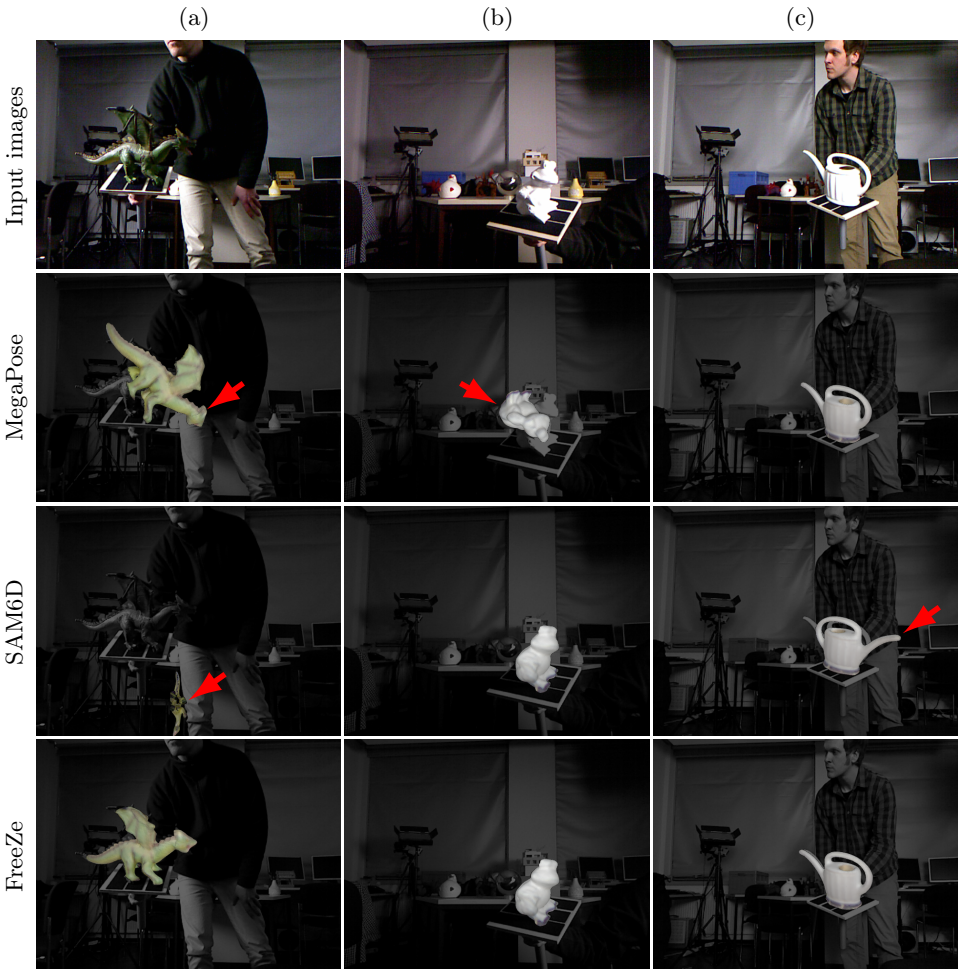

Fig. 7: Qualitative results on the TUD-L [21] dataset. Rows show a comparison against different methods. Columns show different examples. **Red** arrows highlight wrong predictions. Backgrounds are converted to grayscale for a better contrast.

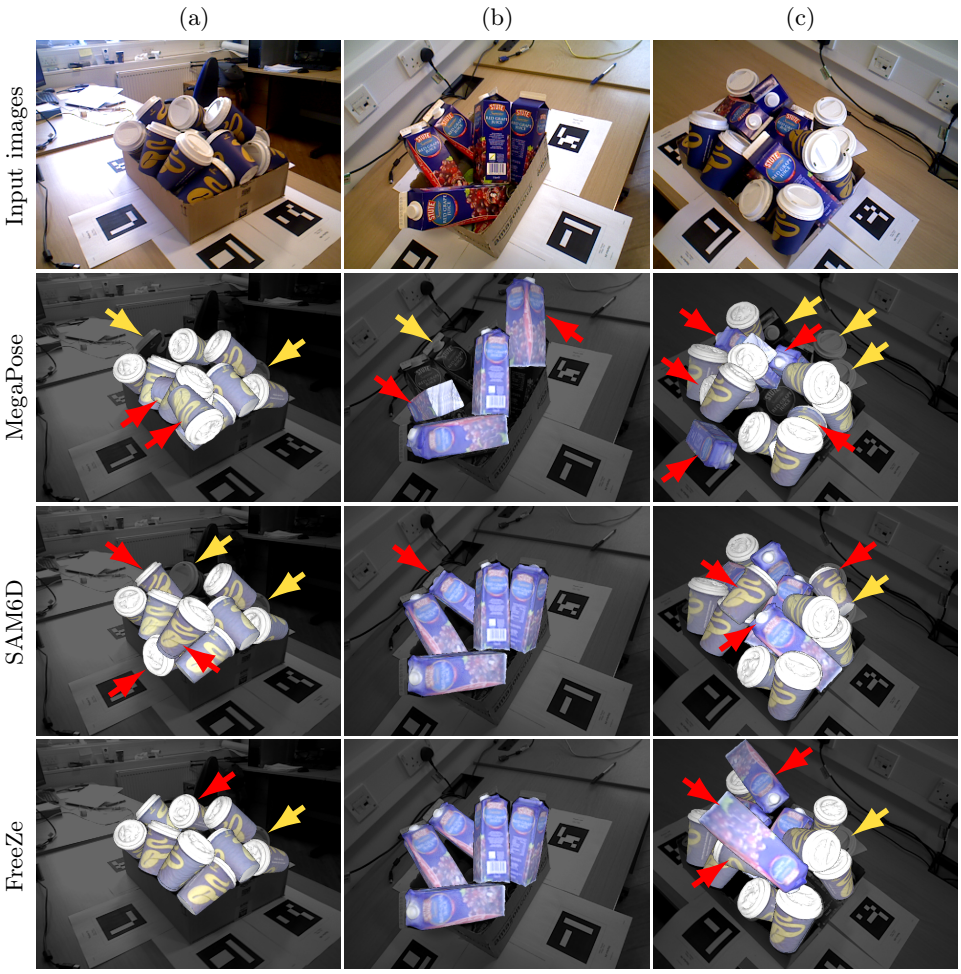

Fig. 8: Qualitative results on IC-BIN [11]. Columns show different examples. Rows show a comparison against different methods. **Red** (yellow) arrows highlight wrong (missing) predictions. Backgrounds are converted to grayscale for a better contrast.

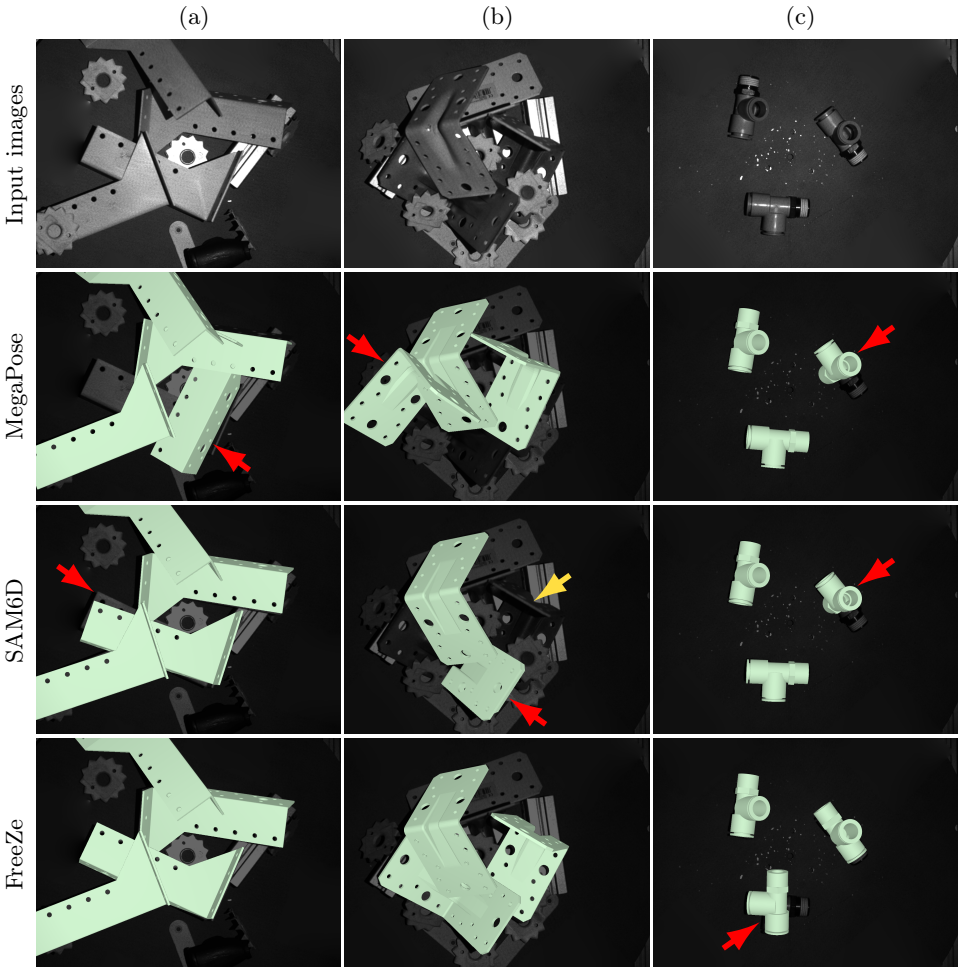

Fig. 9: Qualitative results on ITODD [12]. Columns show different examples. Rows show a comparison against various methods. **Red** (yellow) arrows highlight wrong (missing) predictions. ITODD images are grayscale. The 3D models of texture-less objects are converted to a light-green color for a better contrast.

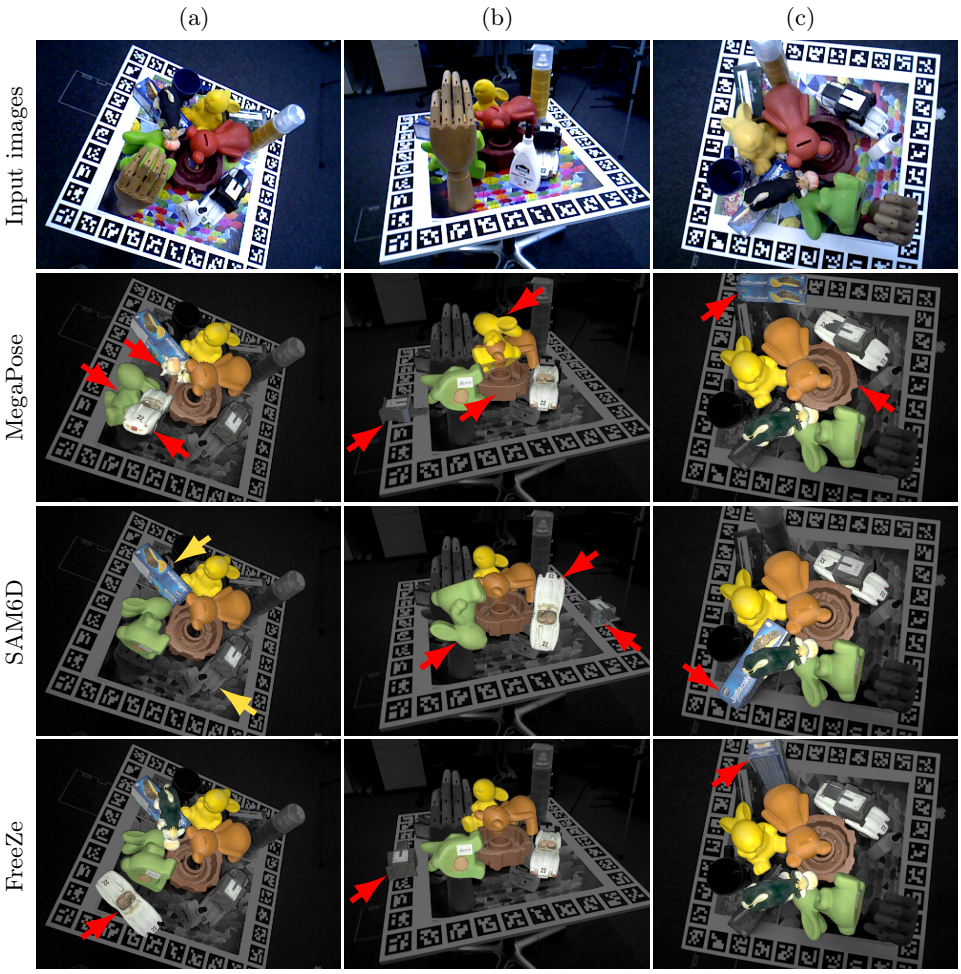

Fig. 10: Qualitative results on HB [24]. Columns show different examples. Rows show a comparison against different methods. **Red** (**yellow**) arrows highlight wrong (missing) predictions. Backgrounds are converted to grayscale for a better contrast.

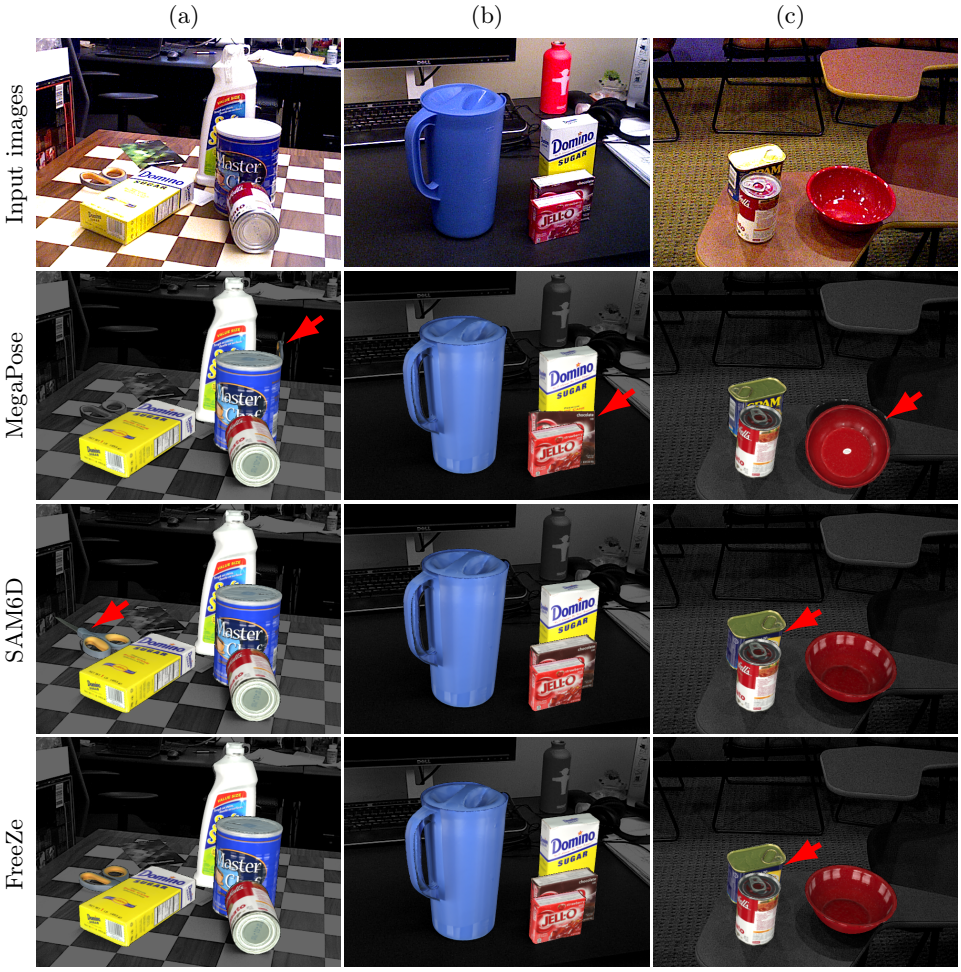

Fig. 11: Qualitative results on YCB-V [53]. Columns show different examples. Rows show a comparison against different methods. **Red** arrows highlight wrong predictions. Backgrounds are converted to grayscale for a better contrast.