

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

Andrea Caraffa¹, Davide Boscaini¹, Amir Hamza^{1,2}, and Fabio Poiesi¹

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 task-specific data. We propose FreeZe, a novel solution that harnesses the capabilities of pre-trained geometric and vision foundation models. FreeZe generates distinctive 3D point-level descriptors by combining 3D geometric descriptors learned from unrelated 3D point cloud datasets and 2D visual features learned from web-scale 2D image collections. We then estimate the 6D pose of unseen objects via RANSAC-based 3D registration. We also introduce a novel algorithm based on visual features to resolve ambiguities caused by geometrically symmetric objects. 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 is 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 the camera’s reference frame. This becomes key for real-world applications such as robotic manipulation [35, 62], augmented reality [50], and autonomous driving [24].

Object 6D pose estimation approaches can be divided into different categories based on the available information about the object of interest. The first categorization is between *model-based* and *model-free* methods: the former require the

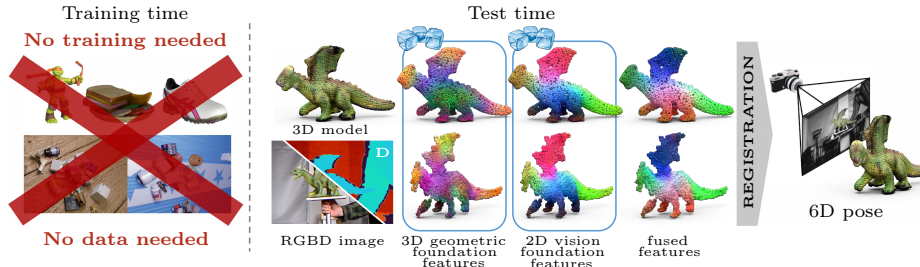


Fig. 1: FreeZe is designed for zero-shot object 6D pose estimation, which involves estimating the pose of an unseen 3D object within a 3D scene (*e.g.* posing the dragon 3D model within the RGBD image displayed in the center). Unlike most competitors, our approach eliminates the need for extensive training 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 two separate frozen pre-trained foundation models.

object’s 3D model as input [3, 6, 37, 53], while the latter relax this assumption and only require a set of reference views [15, 21] or a video [19, 51] of the object. The second distinction lies among *instance-level*, *category-level*, and *zero-shot* methods. Instance-level methods [43, 47, 53] 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 [11, 18, 29, 34] 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, 28, 42] 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 resort to 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 [28] utilizes physically-based rendering techniques to generate a synthetic dataset comprising two million images encompassing 20,000 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 in the foundation models era?* To answer *No!* to this question, we propose a novel training-free zero-shot approach, FreeZe for short, that harnesses the capabilities of pre-trained geometric and vision foundation models without requiring any additional training (Fig. 1). As geometric foundation model we use GeDi [45], which has been trained on 3D point cloud data of indoor scenes, while we use DINOv2 [41] as vision foundation model, which has been trained on web-scale 2D images using 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 scene’s input image using a frozen DINOv2. To compute 2D visual features for the object’s 3D model, we feed DINOv2 with 2D images rendered from different viewpoints around the object. To compute 3D geometric features for the scene, we feed GeDi with a 3D point cloud obtained by 3D-lifting the input depth. During feature fusion, we compute distinctive 3D point-level features by concatenating and normalizing geometric and visual features. The fused features are more discriminative as they encode both 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 case of geometrically symmetric objects, we further refine the pose with a novel Symmetry-Aware Refinement, SAR for short, 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 [22], TUD-L [23], IC-BIN [12], ITODD [13], HB [26], and YCB-V [55]. Together, these datasets encompass over a hundred objects and 20,000 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 to 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 training on task-specific data.

2 Related work

3D deep descriptors are typically used for generic point cloud registration problems [9, 45]. 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 a Local Reference Frame (LRF), which transforms a local set of points (patch) into a canonical representation to achieve rotation and translation invariance [56]. Canonicalized points are then processed by a neural network to produce a compact descriptor as output. Methods in this category include G3DOA [58], which utilizes multi-scale cylindrical convolutions; WSDesc [30], based on voxelization layers that learn the optimal voxel neighborhood size; Li et al. [31], which employs multi-view differentiable rendering; and GeDi [45], which processes canonicalized points with a PointNet++ [46] network. Deep descriptors can also be computed

without relying on a 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 [25], 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 [14, 57]. 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 [48], DINOv2 [41], ImageBind [16], and Segment Anything Model (SAM) [27], can be employed across a range of tasks and domains. CLIP [48] connects visual concepts with textual descriptions using a ViT as image encoder and a GPT-like Transformer as textual encoder, respectively. DINOv2 [41] adopts self-supervised training on large-scale unlabeled data, introducing a patch-level objective to capture fine-grained image details. ImageBind [16] integrates data from six distinct modalities into a shared feature space. This integration is accomplished using only image-paired data, eliminating the need for exhaustive cross-modal combinations. SAM [27] performs object segmentation using visual prompts such as single points, point sets, or bounding boxes. SAM demonstrates strong generalization capabilities, effectively segmenting unseen objects across diverse contexts. Although CLIP has been previously employed for zero-shot point cloud understanding [44, 61], 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 poses of geometrically symmetric objects.

Zero-shot object 6D pose estimation methods, while less numerous compared to fully supervised [20, 32, 53] or few-/one-shot learning approaches [5, 17, 51], are gaining attention. ZePhyR [40] generates a set of candidate poses for the objects and selects the best one based on point-level differences in color and geometry. MegaPose [28] estimates object poses by rendering multiple views of the object’s 3D model and selecting the view that best aligns with the input image using a classifier. ZeroPose [6] uses multi-resolution geometric features to match regions between the scene’s 3D point cloud and the object’s 3D model. Nguyen et al. [39] extracts local features to match the input image against rendered object templates. Similarly, ZS6D [3] matches DINOv2 [41] visual features against a database of features extracted from rendered object templates, and then performs pose estimation using a RANSAC-based PnP algorithm. SAM6D [33] generates mask proposals from images using SAM and ranks them based on a combined score of semantics, appearance, and geometry against rendered object templates. Best 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 [37] 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 [42] leverages a bag-of-words of DINOv2 descriptors to shortlist similar templates. 2D-3D correspondences are then established among query and the selected templates using DINOv2 patch-level features. FoundationPose [54] offers a unified approach for pose estimation and tracking in both model-based and model-free scenarios. The process starts with a coarse pose, 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. The strong generalizability of FoundationPose is achieved through extensive training on large-scale synthetic data augmented via Large Language Models (LLMs).

Unlike existing zero-shot 6D pose estimation approaches, FreeZe utilizes frozen deep neural networks that have been pre-trained on diverse tasks. FreeZe can utilize point cloud descriptors that were trained for point cloud registration, such as GeDi, and image feature extraction networks that were self-supervised on large-scale web data, such as DINOv2. 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 synthetic training data at scale. We take a step forward, and by utilizing recent geometric and vision foundation models, we eliminate the need for any training data at all. Unlike concurrent training-free works, ZS6D and FoundPose, which use visual features to establish 2D-2D and 2D-3D correspondences respectively, we exploit the depth information of the scene to work with 3D-3D correspondences. Moreover, we leverage not only vision foundation models but also geometric ones, demonstrating their synergy.

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 reference frame of the camera that acquired $\mathbf{I}$. Let the *target object* T be the instance of Q captured in $\mathbf{I}$, our goal is to estimate the 6DoF transformation $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ that relates Q to T , where $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ 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 estimating $\mathbf{T}$ from 3D-3D correspondences (Fig. 2). To process Q , we sample N points from the surface of its 3D model to generate a point cloud $\mathcal{P}^Q \in \mathbb{R}^{N \times 3}$ from which we compute point-level geometric features through a frozen pre-trained geometric encoder Ψ . Concurrently, we render templates of Q from different viewpoints, which we feed 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). To process T , we locate it in

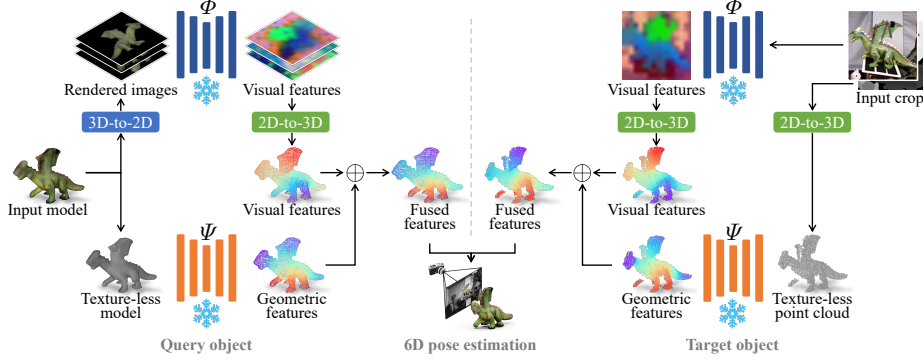


Fig. 2: Overview of FreeZe. Left-hand side: We render templates from different viewpoints around the query object’s 3D model to extract multi-view visual features, which we back-project and aggregate onto the object’s point cloud. Concurrently, we extract geometric features from the object’s point cloud, which we then fuse with the visual features. Right-hand side: We compute visual features (from the RGB channels) and geometric features (from the depth channel) 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 [36]. The symbol $\oplus$ indicates feature concatenation.

I using the segmentation masks estimated by a zero-shot object detector. We generate a point cloud from the pixels belonging to the mask by using $\mathbf{I}$ ’s 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 $\mathbf{I}$ ’s 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 using our novel SAR algorithm (Sec. 3.6).

3.2 Query object processing

We process $\mathcal{P}^Q$ with the geometric encoder Ψ to obtain the geometric features $\mathcal{G}^Q = \{\mathbf{g}_n^Q\}_{n=1}^N = \Psi(\mathcal{P}^Q)$, where $\mathbf{g}_n^Q \in \mathbb{R}^G$ is the G -dimensional feature of the n -th point of Q . 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-level³ visual features $\mathbf{V}_r^Q$ which are then back-projected onto the 3D model, and associated with $\mathcal{P}^Q$ ’s points as

³ We use DINOv2 [41] as Φ , which outputs a 16×16 grid of patch-level features, hence we apply bilinear interpolation to convert patch-level to pixel-level features.

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 [38] or SAM6D [33], to localise T within $\mathbf{I}$. These methods leverage the zero-shot transfer capabilities of SAM [27] (or FastSAM [59]) to generate region proposals within $\mathbf{I}$, and compare them against object templates derived from the textured 3D model of query object. They produce a set of masks with confidence scores based on the visual feature similarity with the object templates. We experimentally observed that these methods are not well calibrated, as more accurate segmentation masks might receive lower confidence scores. Hence, we keep the most confident masks as possible target object candidates. For each candidate, we also extract the smallest bounding box $\mathbf{I}^T \in \mathbf{I}$ containing 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. We compute pixel-level features $\mathbf{V}^T$ from the input crop $\mathbf{I}^T$ with Φ . 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 easily 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, we only transfer foreground features to $\mathcal{P}^T$.

3.4 Geometric and visual feature fusion

We fuse geometric and visual features through concatenation for both Q and T . First, we apply L_2 normalization to $\mathcal{G}^Q$ ($\mathcal{G}^T$) and $\mathcal{V}^Q$ ($\mathcal{V}^T$) independently to account for potential differences in norms and to balance their contribution in the final feature. Specifically, the features we produce have the following form:

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

where $\mathbf{f}_n^Q, \mathbf{f}_m^T \in \mathbb{R}^{G+V}$, $[\cdot]$ denotes concatenation, and $\|\cdot\|_2$ the L_2 norm operator. We found this simple fusion strategy to work well in practice. However, alternative fusion strategies could also be effective, and we leave this for future exploration.

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 the one with the highest number of registration inliers.

3.6 Symmetry-aware refinement

We use the Iterative Closest Point (ICP) algorithm [7] to refine $\mathbf{T}_c$ at point level and obtain a finer transformation $\mathbf{T}_f$. Then, we use visual features computed by Φ to solve pose ambiguities that may arise from geometric symmetries. Specifically, we address the problem by introducing SAR, a novel pose refinement based on rendering and visual features matching. SAR aims to adjust the pose of geometrically symmetric objects for which the estimated pose would be correct without considering the texture, *i.e.* when $\mathcal{P}^Q$ and $\mathcal{P}^T$ match, but their textures do not. SAR incorporates photometric information through visual features, ensuring optimal solutions both geometrically and photometrically.

Symmetry estimation. We estimate the object’s geometric symmetries, *i.e.* regardless of texture. We build a set of rotation matrices by uniformly sampling the rotation space. The estimated symmetries are the set of rotations $\{\mathbf{R}_s \in SO(3)\}^S$ that minimize the Chamfer distance [60] between $\mathbf{R}_s \mathcal{P}^Q$ and $\mathcal{P}^Q$.

Symmetry selection. We render the RGB images $\{\mathbf{I}_s\}_{s=1}^S$ depicting the 3D model of Q in the poses $\{\mathbf{T}_f \circ \mathbf{R}_s\}_{s=1}^S$. We use the bounding boxes defined in Sec. 3.3 to crop images $\{\mathbf{I}_s^T\}_{s=1}^S$. We then process $\{\mathbf{I}_s^T\}_{s=1}^S$ with Φ to produce visual features $\{\mathbf{V}_s\}_{s=1}^S$. The best candidate is selected by computing the cosine similarity between $\{\mathbf{V}_s\}_{s=1}^S$ 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 [38]. Moreover, as both visual features coming from $\mathbf{I}^T$ and $\mathbf{I}_s$ 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 SAR. Starting from an inaccurately estimated pose, we render images of the object posed according to its estimated symmetries. We compute the patch-level cosine similarity between the visual features of the input crop and the rendered images. Symmetry no. 3 exhibits a higher similarity score (+30%), enabling us to correct the initial pose.

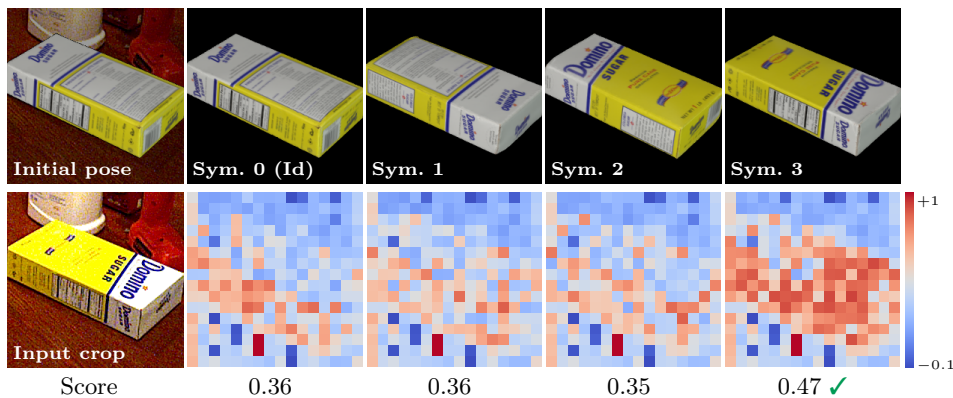


Fig. 3: Example of Symmetry-Aware Refinement (SAR). We adjust the initial pose prediction (top left corner) by comparing the input crop (bottom left corner) with images rendered considering the estimated object’s symmetries (first row). Sym. 0 (Id) represents the identity transformation and its rendered image is obtained considering only the initial pose. 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 finally select the best candidate as the one with the highest score (✓).

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 segmentation masks estimated with CNOS [38] and SAM6D [33] to localize the object within the input image. As geometric encoder we use GeDi [45], 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 [41]. DINOv2 is a vision foundation model trained through self-supervision on large-scale web data. DINOv2 processes images of size 224×224 , and its ViT-giant version outputs 1536-dimensional features. We then apply PCA to reduce their dimensionality to $V = 64$, for both efficiency and to match dimensionality of geometric features.

4.2 Datasets

We evaluate FreeZe on the seven core datasets of the BOP Benchmark [52]: LM-O [4], T-LESS [22], TUD-L [23], IC-BIN [12], ITODD [13], HB [26], and YCB-V [55]. Each dataset provides both 3D models 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 on average 4 instances per object, except a few images that contain 84 instances per object. T-LESS and IC-BIN contain on average 1.3 and 9 instances per object, 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 per 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 [52], which are VSD, MSSD, and MSPD. These metrics assess different errors between the two models obtained by applying the predicted and ground-truth poses. VSD measures the discrepancy between the depth maps after rendering. 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 the 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 reported methods use a localization prior estimated in a zero-shot fashion, *i.e.* for a fair comparison, we only report methods that operate in the same setting as ours. The table is divided into two sections: the top part lists methods with CNOS localization priors, while the bottom part assumes SAM6D localization priors. For completeness, the table also includes RGB-only methods, although they are not directly comparable. 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 [33] (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 [28] (row 11) and a +13.6% with respect to SAM6D (row 12). TUD-L is renowned as a dataset featuring exceptionally challenging lighting conditions. In this context, FreeZe stands out

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. All the results are achieved with a zero-shot localization prior, we do not include any result with supervised or undefined localization 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							Mean
					LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	
1 MegaPose [28]		RGB			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 [37]		RGB			29.9	27.3	30.2	23.1	18.8	34.8	29.0	27.6
4 FoundPose [42]	✓	RGB	CNOS		39.7	33.8	46.9	23.9	20.4	50.8	45.2	37.3
5 SAM6D [33]		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 [28]		RGB		✓	56.0	50.8	68.7	41.9	34.6	70.6	62.0	54.9
8 GigaPose [37]		RGB		✓	59.9	57.0	63.5	46.7	39.7	72.2	66.3	57.9
9 FoundPose [42]	✓	RGB		✓	61.0	57.0	69.3	47.9	40.7	72.3	69.0	59.6
10 ZeroPose [6]		RGBD	CNOS	✓	53.8	40.0	83.5	39.2	52.1	65.3	65.3	57.0
11 MegaPose [28]		RGBD		✓	62.6	48.7	85.1	46.7	46.8	73.0	76.4	62.8
12 SAM6D [33]		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 [33]		RGBD SAM6D			62.7	42.0	77.7	50.4	45.5	68.9	74.3	60.2
15 FreeZe (ours)	✓	RGBD SAM6D			67.6	50.0	88.1	48.7	52.0	76.1	77.4	65.7
16 SAM6D [33]		RGBD SAM6D		✓	68.7	49.8	87.4	56.1	57.7	75.4	82.8	68.2
17 FreeZe (ours)	✓	RGBD SAM6D		✓	71.6	53.1	94.9	54.5	58.6	79.6	84.0	70.9

by demonstrating the remarkable efficacy of leveraging geometric features to their fullest extent. Remarkably, 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 refinement. Finally, FreeZe using CNOS masks outperforms SAM6D using SAM6D masks by +1% (row 13 vs row 16), which is remarkable since SAM6D masks are more accurate.

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, similarity with other objects and 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

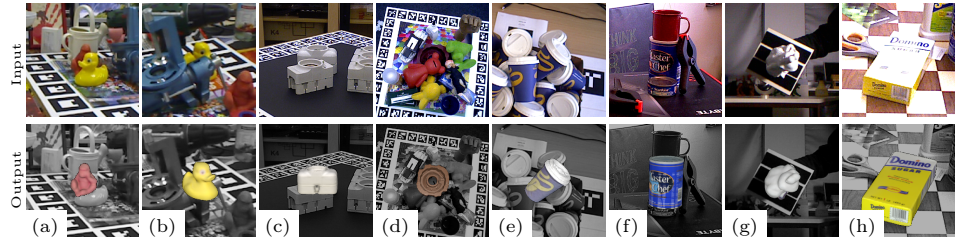


Fig. 4: Qualitative results of challenging cases. The top row shows input images, while the bottom row shows FreeZe’s predictions by overlaying the object 3D model transformed according to the predicted pose. We use a grayscale version of the input image for a better contrast. Left to right: (a,b) highly-occluded 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.

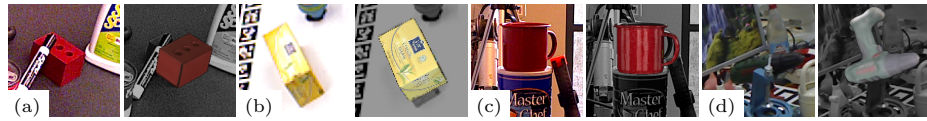


Fig. 5: Failure cases. FreeZe’s predictions follow the relative input images. Left to right: (a) the foam block is aligned only to the closest corner, in a pose that is almost perpendicular to the correct one, (b) the tea box pose is predicted as if it was laying on its side rather than on its bottom, (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.

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.

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 CNOS segmentation masks as localization prior across all the experiments, unless otherwise specified.

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 [49], and learning-based techniques FCGF [9] and GeDi [45]. In rows 9–11 of Tab. 2, we evaluate different vision encoders: simple RGB values, and the foundation models CLIP [48] and DINOv2 [41]. 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 [45] uses a single radius to define the extent of local descriptors. Instead, we propose to 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 bounding boxes as localization prior as opposite to segmentation masks. Although masks may offer higher precision, they can be noisy and fail to capture all object parts, whereas bounding boxes cover larger portions more reliably. The table demonstrates that FreeZe without refinement achieves equivalent AR regardless of the localization prior (rows 16 and 18), underscoring its robustness across background variations. Instead, once we refine the pose using ICP, FreeZe performs better with the segmentation masks, suggesting that ICP is more sensitive to background outliers.

Pose refinement. In Tab. 3, we evaluate the effectiveness of the pose refinement module by comparing the coarse estimation (row 1) with the pose obtained using a vanilla ICP-based refinement (row 2), and the one obtained adding also our novel SAR algorithm (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). Notably, SAR only marginally improves the results, as most of the time the visual features already enable FreeZe to find relevant correspondences.

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 step, with a vanilla 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

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. A solution to mitigate this problem is by employing distillation techniques. Other **future research** directions include improving our 3D registration process, possibly by advancing beyond simple RANSAC algorithms, and predicting geometric features from RGB-only data.

Acknowledgements

This work was supported by the European Union’s Horizon Europe research and innovation programme under grant agreement No. 101058589 (AI-PRISM). This study was also carried out within by the Interconnected Nord-Est Innovation Ecosystem (iNEST) and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.5 – D.D. 1058 23/06/2022, ECS00000043).

References

1. 6D localization of unseen objects – Core datasets. <https://bop.felk.cvut.cz/leaderboards/pose-estimation-unseen-bop23/core-datasets/>, 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., Habegger, 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. In: ICCV-W (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. Di Felice, F., Remus, A., Gasperini, S., Busam, B., Ott, L., Tombari, F., Siegwart, R., Avizzano, C.A.: Zero123-6D: Zero-shot novel view synthesis for rgb category-level 6D pose estimation. arXiv:2403.14279 (2024)
12. Dumanoglou, A., Kouskouridas, R., Malassiotis, S., Kim, T.K.: Recovering 6D object pose and predicting next-best-view in the crowd. In: CVPR (2016)
13. 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)
14. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3D object recognition. In: CVPR (2010)
15. 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)
16. 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)
17. 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)

18. Goodwin, W., Vaze, S., Havoutis, I., Posner, I.: Zero-shot category-level object pose estimation. In: ECCV (2022)
19. 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)
20. He, Y., Huang, H., Fan, H., Chen, Q., Sun, J.: FFB6D: A full flow bidirectional fusion network for 6D pose estimation. In: CVPR (2021)
21. He, Y., Wang, Y., Fan, H., Sun, J., Chen, Q.: FS6D: Few-shot 6D pose estimation of novel objects. In: CVPR (2022)
22. 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)
23. 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)
24. 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)
25. Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., Schindler, K.: PREDATOR: Registration of 3D point clouds with low overlap. In: CVPR (2021)
26. Kaskman, R., Zakharov, S., Shugurov, I., Ilic, S.: HomebrewedDB: RGB-D dataset for 6D pose estimation of 3D objects. In: ICCV-W (2019)
27. 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)
28. 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)
29. 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)
30. 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)
31. 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)
32. 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)
33. Lin, J., Liu, L., Lu, D., Jia, K.: SAM-6D: Segment anything model meets zero-shot 6D object pose estimation. arXiv:2311.15707 (2023)
34. 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)
35. 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)
36. van der Maaten, L., Hinton, G.: Visualizing high-dimensional data using t-SNE. Journal of Machine Learning Research (2008)
37. Nguyen, V.N., Groueix, T., Salzmann, M., Lepetit, V.: GigaPose: Fast and robust novel object pose estimation via one correspondence. arXiv:2311.14155 (2023)
38. Nguyen, V.N., Hodan, T., Ponimatin, G., Groueix, T., Lepetit, V.: CNOS: A strong baseline for CAD-based novel object segmentation. In: ICCV-W (2023)

39. 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: CVPR (2022)
40. Okorn, B., Gu, Q., Hebert, M., Held, D.: ZePHyR: Zero-shot pose hypothesis rating. In: ICRA (2021)
41. 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)
42. Örneke, E.P., Labbé, Y., Tekin, B., Ma, L., Keskin, C., Forster, C., Hodan, T.: Found-Pose: Unseen object pose estimation with foundation features. arXiv:2311.18809 (2023)
43. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: PVNet: Pixel-wise voting network for 6DoF pose estimation. In: CVPR (2019)
44. Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T., et al.: OpenScene: 3D scene understanding with open vocabularies. In: CVPR (2023)
45. Poiesi, F., Boscaini, D.: Learning general and distinctive 3D local deep descriptors for point cloud registration. IEEE TPAMI (2023)
46. 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)
47. Rad, M., Oberweger, M., Lepetit, V.: Feature mapping for learning fast and accurate 3D pose inference from synthetic images. In: CVPR (2018)
48. 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: ICML (2021)
49. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: ICRA (2009)
50. 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)
51. 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)
52. 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)
53. 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)
54. Wen, B., Yang, W., Kautz, J., Birchfield, S.: FoundationPose: Unified 6D pose estimation and tracking of novel objects. In: CVPR (2024)
55. 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)
56. Yang, J., Zhang, Q., Xiao, Y., Cao, Z.G.: TOLDI: An effective and robust approach for 3D local shape description. Pattern Recognition (2017)
57. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. In: CVPR (2017)
58. Zhao, H., Zhuang, H., Wang, C., Yang, M.: G3DOA: Generalizable 3D descriptor with overlap attention for point cloud registration. RA-L (2022)
59. Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M., Tang, M., Wang, J.: Fast segment anything. arXiv:2306.12156 (2023)

- 60. Zhao, Y., Birdal, T., Deng, H., Tombari, F.: 3D point capsule networks. In: CVPR (2019)
- 61. 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)
- 62. 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)