FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects

Bowen Wen Wei Yang Jan Kautz Stan Birchfield

NVIDIA

Abstract

We present FoundationPose, a unified foundation model for 6D object pose estimation and tracking, supporting both model-based and model-free setups. Our approach can be instantly applied at test-time to a novel object without fine-tuning, as long as its CAD model is given, or a small number of reference images are captured. We bridge the gap between these two setups with a neural implicit representation that allows for effective novel view synthesis, keeping the downstream pose estimation modules invariant under the same unified framework. Strong generalizability is achieved via large-scale synthetic training, aided by a large language model (LLM), a novel transformer-based architecture, and contrastive learning formulation. Extensive evaluation on multiple public datasets involving challenging scenarios and objects indicate our unified approach outperforms existing methods specialized for each task by a large margin. In addition, it even achieves comparable results to instance-level methods despite the reduced assumptions. Project page: https://nvlabs.github.io/FoundationPose/

1. Introduction

Computing the rigid 6D transformation from the object to the camera, also known as object pose estimation, is crucial for a variety of applications, such as robotic manipulation [28, 60] and mixed reality [39]. Classic methods [18, 19, 29, 45] are known as *instance-level* because they only work on the specific object instance determined at training time. Such methods usually require a textured CAD model for generating training data, and they cannot be applied to an unseen novel object at test time. While *category-level methods* [5, 31, 52, 56, 65] remove these assumptions (instance-wise training and CAD models), they are limited to objects within the predefined category on which they are trained. Moreover, obtaining category-level training data is notoriously difficult, in part due to additional pose canonicalization and examination steps [56] that must be applied.

To address these limitations, more recent efforts have focused on the problem of instant pose estimation of arbitrary novel objects [17, 30, 37, 48, 51]. Two different setups are considered, depending upon what information is available at test time: *model-based*, where a textured 3D CAD model of the object is provided, and *model-free*, where a set of reference images of the object is provided. While much progress has been made on both setups individually, there remains a need for a single method to address both setups in a unified way, since different real-world applications provide different types of information.

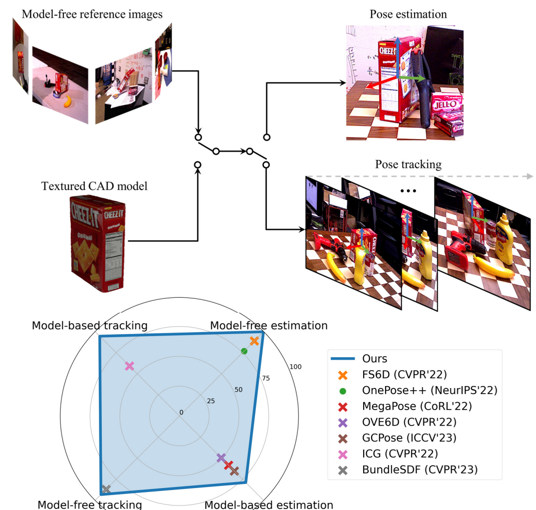

Figure 1. Our unified framework enables both 6D pose estimation and tracking for novel objects, supporting the model-based and model-free setups. On each of these four tasks, it outperforms prior work specially designed for the task (• indicates RGB-only, × indicates RGBD). The metric for each task is explained in detail in the experimental results.

Orthogonal to single-frame object pose estimation, pose tracking methods [8, 27, 33, 36, 49, 55, 59, 62] leverage temporal cues to enable more efficient, smooth and accurate pose estimation on a video sequence. These methods share the similar aforementioned issues to their counterparts in pose estimation, depending on their assumptions on the object knowledge.

In this paper we propose a unified framework called *FoundationPose* that performs both pose estimation and tracking for novel objects in both the model-based and model-free setups. As seen in Fig. 1, our method outperforms existing state-of-art methods specialized for each of these four tasks. Our strong generalizability is achieved via large-scale synthetic training, aided by a large language model (LLM), as well as a novel transformer-based architecture and contrastive learning. We bridge the gap between model-based and model-free setups with a neural implicit representation that allows for effective novel view synthesis with a small number ($\sim$16) of reference images, achieving rendering speeds that are significantly faster than previous render-and-compare methods [30, 33, 59]. Our contributions can be summarized as follows:

- We present a unified framework for both pose estimation and tracking for novel objects, supporting both model-based and model-free setups. An object-centric neural implicit representation for effective novel view synthesis bridges the gap between the two setups.
- We propose a LLM-aided synthetic data generation pipeline which scales up the variety of 3D training assets by diverse texture augmentation.
- Our novel design of transformer-based network architectures and contrastive learning formulation leads to strong generalization when trained solely on synthetic data.
- Our method outperforms existing methods specialized for each task by a large margin across multiple public datasets. It even achieves comparable results to instance-level methods despite reduced assumptions.

Code and data developed in this work will be released.

2. Related Work

CAD Model-based Object Pose Estimation. Instance-level pose estimation methods [18, 19, 29, 45] assume a textured CAD model is given for the object. Training and testing is performed on the exact same instance. The object pose is often solved by direct regression [34, 63], or constructing 2D-3D correspondences followed by PnP [45, 53], or 3D-3D correspondences followed by least squares fitting [18, 19]. To relax the assumptions about the object knowledge, category-level methods [5, 31, 52, 56, 65, 67] can be applied to novel object instances of the same category, but they cannot generalize to arbitrary novel objects beyond the predefined categories. To address this limitation, recent efforts [30, 48] aim for instant pose estimation of arbitrary novel objects as long as the CAD model is provided at test time.

Few-shot Model-free Object pose estimation. Model-free methods remove the requirement of an explicit textured model. Instead, a number of reference images capturing the target object are provided [17, 20, 46, 51]. RLLG [3] and NeRF-Pose [32] propose instance-wise training without the need of an object CAD model. In particular, [32] constructs a neural radiance field to provide semi-supervision on the object coordinate map and mask. Differently, we introduce the neural object field built on top of SDF representation for efficient RGB and depth rendering to bridge the gap between the model-based and model-free scenarios. In addition, we focus on generalizable novel object pose estimation in this work, which is not the case for [3, 32]. To handle novel objects, Gen6D [37] designs a detection, retrieval and refinement pipeline. However, to avoid difficulties with out-of-distribution test set, it requires fine-tuning. OnePose [51] and its extension OnePose++ [17] leverage structure-from-motion (SfM) for object modeling and pretrain 2D-3D matching networks to solve the pose from correspondences. FS6D [20] adopts a similar scheme and focuses on RGBD modality. Nevertheless, reliance on correspondences becomes fragile when applied to textureless objects or under severe occlusion.

Object Pose Tracking. 6D object pose tracking aims to leverage temporal cues to enable more efficient, smooth and accurate pose prediction on video sequence. Through neural rendering, our method can be trivially extended to the pose tracking task with high efficiency. Similar to single-frame pose estimation, existing tracking methods can be categorized into their counterparts depending on the assumptions of object knowledge. These include instance-level methods [8, 33, 59], category-level methods [36, 55], model-based novel object tracking [27, 49, 62] and model-free novel object tracking [58, 61]. Under both model-based and model-free setups, we set a new benchmark record across public datasets, even outperforming state-of-art methods that require instance-level training [8, 33, 59].

3. Approach

Our method is described in the following subsections. The relationships between the subsections, and the system as a whole, are illustrated in Fig. 2.

3.1. Language-aided Data Generation at Scale

To achieve strong generalization, a large diversity of objects and scenes is needed for training. Obtaining such data in the real world, and annotating accurate ground-truth 6D pose, is time- and cost-prohibitive. Synthetic data, on the other hand, often lacks the size and diversity in 3D assets. We developed a novel synthetic data generation pipeline for training, powered by the recent emerging resources and techniques: large scale 3D model database [6, 10], large language models (LLM), and diffusion models [4, 22, 47]. This approach dramatically scales up both the amount and diversity of data compared with prior work [20, 24, 30].

3D Assets. We obtain training assets from recent large scale 3D databases including Objaverse [6] and GSO [10]. For Objaverse [6] we chose the objects from the Objaverse-LVIS subset that consists of more than 40K objects belonging to 1156 LVIS [12] categories. This list contains the most relevant daily-life objects with reasonable quality, and diversity of shapes and appearances. It also provides a tag for each object describing its category, which benefits au-

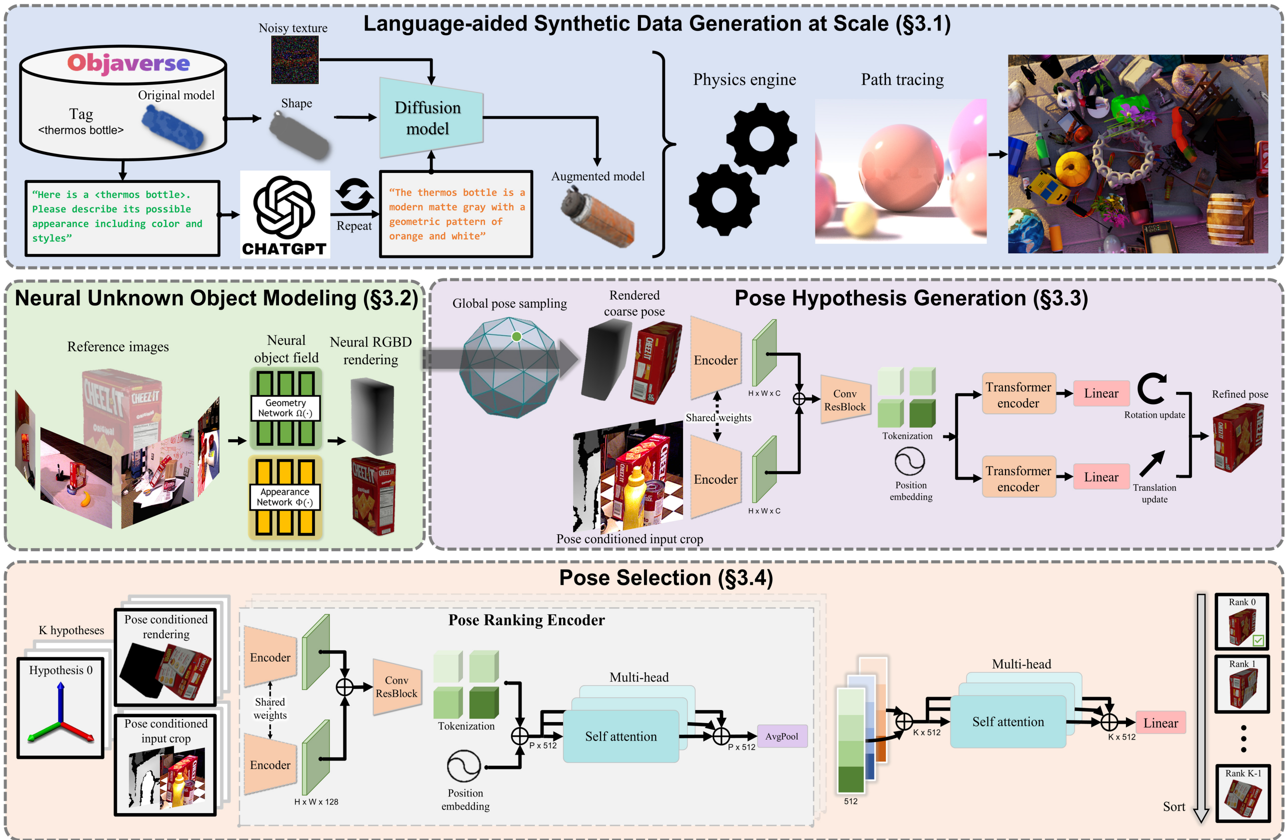

Figure 2. Overview of our framework. To reduce manual efforts for large scale training, we developed a novel synthetic data generation pipeline by leveraging recent emerging techniques and resources including 3D model database, large language models and diffusion models (Sec. 3.1). To bridge the gap between model-free and model-based setup, we leverage an object-centric neural field (Sec. 3.2) for novel view RGBD rendering for subsequent render-and-compare. For pose estimation, we first initialize global poses uniformly around the object, which are then refined by the refinement network (Sec. 3.3). Finally, we forward the refined poses to the pose selection module which predicts their scores. The pose with the best score is selected as output (Sec. 3.4).

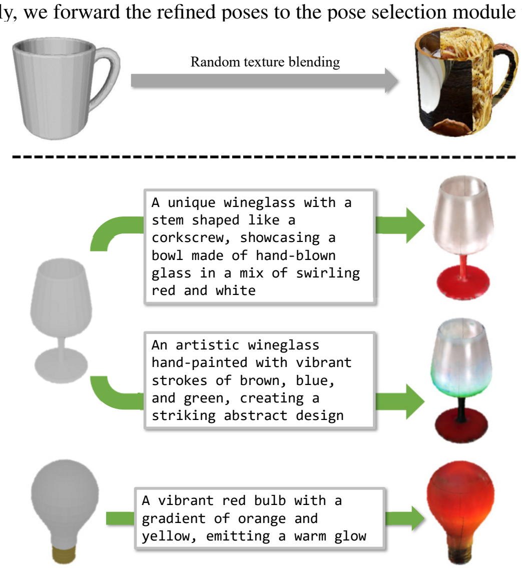

Figure 3. **Top:** Random texture blending proposed in FS6D [20]. **Bottom:** Our LLM-aided texture augmentation yields more realistic appearance. Leftmost is the original 3D assets. Text prompts are automatically generated by ChatGPT.

tomatic language prompt generation in the following LLM-aided texture augmentation step.

LLM-aided Texture Augmentation. While most Objaverse objects have high quality shapes, their texture fidelity varies significantly. FS6D [20] proposes to augment object texture by randomly pasting images from ImageNet [7] or MS-COCO [35]. However, due to the random UV mapping, this method yields artifacts such as seams on the resulting textured mesh (Fig. 3 top); and applying holistic scene images to objects leads to unrealistic results. In contrast, we explore how recent advances in large language models and diffusion models can be harnessed for more realistic (and fully automatic) texture augmentation. Specifically, we provide a text prompt, an object shape, and a randomly initialized noisy texture to TexFusion [4] to produce an augmented textured model. Of course, providing such a prompt manually is not scalable if we want to augment a large number of objects in diverse styles under different prompt guidance. As a result, we introduce a two-level hierarchical prompt strategy. As illustrated in Fig. 2 top-left, we first prompt ChatGPT, asking it to describe the possible appearance of an object; this prompt is templated so that each time we only need to replace the tag paired with the object, which is given by the Objaverse-LVIS list. The answer from ChatGPT then becomes the text prompt provided to the diffusion model for texture synthesis. Because this approach enables full automation for texture augmentation, it facilitates diversified data generation at scale. Fig. 3 presents more examples including different stylization for the same object.

Data Generation. Our synthetic data generation is implemented in NVIDIA Isaac Sim, leveraging path tracing for high-fidelity photo-realistic rendering.[1] We perform gravity and physics simulation to produce physically plausible scenes. In each scene, we randomly sample objects including the original and texture-augmented versions. The object size, material, camera pose, and lighting are also randomized; more details can be found in the appendix.

3.2. Neural Unknown Object Modeling

In the absence of CAD models, one key challenge is to represent the object to effectively render images with sufficient quality for downstream modules. Neural implicit representations have not only been shown to be effective for novel view synthesis, but they can also be parallelized on a GPU, thus providing high computational efficiency when rendering multiple pose hypotheses for downstream pose estimation modules. To this end, we introduce an object-centric neural SDF representation for object modeling, inspired by previous work [41, 57, 61, 64].

Field Representation. We represent the object by two functions [64] as shown in Fig. 2. First, the geometry function $\Omega : x \mapsto s$ takes as input a 3D point $x \in \mathbb{R}^3$ and outputs a signed distance value $s \in \mathbb{R}$. Second, the appearance function $\Phi : (f_{\Omega(x)}, n, d) \mapsto c$ takes the intermediate feature vector $f_{\Omega(x)}$ from the geometry network, a point normal $n \in \mathbb{R}^3$, and a view direction $d \in \mathbb{R}^3$, and outputs the color $c \in \mathbb{R}^3_+$. In practice, we apply multi-resolution hash encoding [41] to x before forwarding to the network. Both n and d are embedded by a fixed set of second-order spherical harmonic coefficients. The implicit object surface is obtained by taking the zero level set of the signed distance field: $S = \left\{x \in \mathbb{R}^3 \mid \Omega(x) = 0\right\}$. Compared to NeRF [40], the SDF representation Ω provides higher quality depth rendering while removing the need to manually select a density threshold.

Field Learning. For texture learning, we follow the volumetric rendering over truncated near-surface regions [61]:

$$c(r) = \int_{z(r)-\lambda}^{z(r)+0.5\lambda} w(x_i)\Phi(f_{\Omega(x_i)}, n(x_i), d(x_i))\, dt, \quad (1)$$

$$w(x_i) = \frac{1}{1+e^{-\alpha\Omega(x_i)}} \frac{1}{1+e^{\alpha\Omega(x_i)}}, \quad (2)$$

where $w(x_i)$ is the bell-shaped probability density function [57] that depends on the signed distance $\Omega(x_i)$ from the point to the implicit object surface, and α adjusts the softness of the distribution. The probability peaks at the surface intersection. In Eq. (1), $z(r)$ is the depth value of the ray from the depth image, and λ is the truncation distance. We ignore the contribution from empty space that is more than λ away from the surface for more efficient training, and we only integrate up to a 0.5λ penetrating distance to model self-occlusion [57]. During training, we compare this quantity against the reference RGB images for color supervision:

$$\mathcal{L}_c = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \left\| c(r) - \bar{c}(r) \right\|_2, \quad (3)$$

where $\bar{c}(r)$ denotes the ground-truth color at the pixel where the ray r passes through.

For geometry learning, we adopt the hybrid SDF model [61] by dividing the space into two regions to learn the SDF, leading to the empty space loss and the near-surface loss. We also apply Eikonal regularization [11] to the near-surface SDF:

$$\mathcal{L}_e = \frac{1}{|\mathcal{X}_e|} \sum_{x \in \mathcal{X}_e} |\Omega(x) - \lambda|, \quad (4)$$

$$\mathcal{L}_s = \frac{1}{|\mathcal{X}_s|} \sum_{x \in \mathcal{X}_s} \left(\Omega(x) + d_x - d_D\right)^2, \quad (5)$$

$$\mathcal{L}_{eik} = \frac{1}{|\mathcal{X}_s|} \sum_{x \in \mathcal{X}_s} (\left\|\nabla\Omega(x)\right\|_2 - 1)^2, \quad (6)$$

where x denotes a sampled 3D point along the rays in the divided space; d_x and d_D are the distance from ray origin to the sample point and the observed depth point, respectively. We do not use the uncertain free-space loss [61], as the template images are pre-captured offline in the model-free setup. The total training loss is

$$\mathcal{L} = w_c\mathcal{L}_c + w_e\mathcal{L}_e + w_s\mathcal{L}_s + w_{eik}\mathcal{L}_{eik}. \quad (7)$$

The learning is optimized per object without priors and can be efficiently performed within seconds. When training the pose refinement (Sec. 3.3) and selection (Sec. 3.4) modules, we first pretrain the neural object field with randomized number of synthetic reference images capturing the 3D asset. The trained neural object field then provides rendering which will be mixed with the model-based OpenGL rendering as input to subsequent networks. This better covers the distribution of both model-based and model-free setups, enabling effective generalization as a unified framework.

Rendering. The Neural Field only needs to be trained once for a novel unknown object. Once trained, the field is efficiently rendered at inference, serving as a drop-in replacement for a conventional graphics pipeline. In addition to the color rendering as in the original NeRF [40], we also need depth rendering for our RGBD based pose estimation and tracking. To do so, we perform marching cubes [38] to extract a mesh from the zero level set of the SDF. This only needs to be performed once for each object. At inference, given an object pose, we then render the depth image following the rasterization process. Alternatively, one could

[1] https://developer.nvidia.com/isaac-sim

directly render the depth image using Ω online with sphere tracing [13]; however, we found this leads to less efficiency, especially when there is a large number of pose hypotheses to render in parallel.

3.3. Pose Hypothesis Generation

Pose Initialization. Given the RGBD image, we first initialize the translation using the 3D point located at the median depth within the region of interest defined by the 2D detection. To initialize rotations, we uniformly sample N_s viewpoints from an icosphere centered on the object with the camera facing the center. These camera poses are further augmented with N_i discretized in-plane rotations, resulting in $N_s \cdot N_i$ global pose initializations which are sent as input to the pose refiner.

Pose Refinement. Since the coarse pose initializations from the previous step are often quite noisy, a refinement module is needed to improve the pose quality. Specifically, we build a pose refinement network which takes as input the rendering of the object conditioned on the coarse pose, and a crop of the input observation from the camera; the network outputs a pose update that improves the pose quality. Unlike MegaPose [30], which renders multiple views around the coarse pose to find the anchor point, we observed rendering a single view corresponding to the coarse pose suffices. For the input observation, instead of cropping based on the 2D detection which is constant, we perform a pose-conditioned cropping strategy so as to provide feedback to the translation update. Concretely, we project the object origin to the image space to determine the crop center. We then project the slightly enlarged object diameter (the maximum distance between any pair of points on the object surface) to determine the crop size that encloses the object and the nearby context around the pose hypothesis. This crop is thus conditioned on the coarse pose and encourages the network to update the translation to make the crop better aligned with the observation. The refinement process can be repeated multiple times by feeding the latest updated pose as input to the next inference, so as to iteratively improve the pose quality.

The refinement network architecture is illustrated in Fig. 2; details are in the appendix. We first extract feature maps from the two RGBD input branches with a single shared CNN encoder. The feature maps are concatenated, fed into CNN blocks with residual connection [15], and tokenized by dividing into patches [9] with position embedding. Finally, the network predicts the translation update $\Delta\boldsymbol{t} \in \mathbb{R}^3$ and rotation update $\Delta\boldsymbol{R} \in \mathbb{SO}(3)$, each individually processed by a transformer encoder [54] and linearly projected to the output dimension. More concretely, $\Delta\boldsymbol{t}$ represents the object's translation shift in the camera frame, $\Delta\boldsymbol{R}$ represents the object's orientation update expressed in the camera frame. In practice, the rotations are parameterized with the 6D representation [68]. The input coarse pose $[\boldsymbol{R}\,|\,\boldsymbol{t}] \in \mathbb{SE}(3)$ is then updated by:

$$\boldsymbol{t}^+ = \boldsymbol{t} + \Delta\boldsymbol{t} \tag{8}$$

$$\boldsymbol{R}^+ = \Delta\boldsymbol{R} \otimes \boldsymbol{R}, \tag{9}$$

where $\otimes$ denotes update on $\mathbb{SO}(3)$. Instead of using a single homogeneous pose update, this disentangled representation removes the dependency on the updated orientation when applying the translation update. This unifies both the updates and input observation in the camera coordinate frame and thus simplifies the learning process. The network training is supervised by L_2 loss:

$$\mathcal{L}_{\text{refine}} = w_1 \left\|\Delta\boldsymbol{t} - \Delta\bar{\boldsymbol{t}}\right\|_2 + w_2 \left\|\Delta\boldsymbol{R} - \Delta\bar{\boldsymbol{R}}\right\|_2, \tag{10}$$

where $\bar{\boldsymbol{t}}$ and $\bar{\boldsymbol{R}}$ are ground truth.

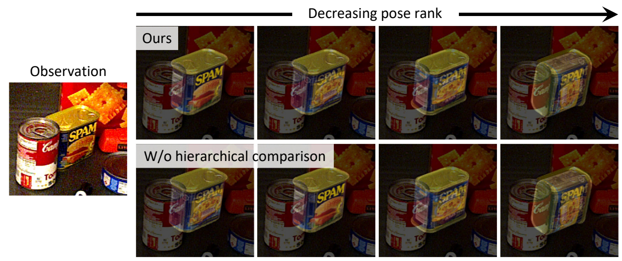

Figure 4. Pose ranking visualization. Our proposed hierarchical comparison leverages the global context among all pose hypotheses for a better overall trend prediction that aligns both shape and texture.

3.4. Pose Selection

Given a list of refined pose hypotheses, we build a pose ranking network to compute their scores. The pose with the highest score is selected as the final estimate.

Hierarchical Comparison. We propose a two-level comparison strategy. First, for each pose hypothesis, we compare the rendered image against the cropped input observation, where the pose-conditioned cropping operation was introduced in Sec. 3.3. This comparison is performed with a pose ranking encoder, where we utilize the same backbone architecture for feature extraction as in the refinement network. The extracted features are concatenated, tokenized and forwarded to the multi-head self-attention module so as to better leverage the global context for comparison. The pose ranking encoder performs average pooling to output a feature embedding $\mathcal{F} \in \mathbb{R}^{512}$ describing the alignment quality between the rendering and the observation (Fig. 2 bottom-middle). At this point, we could directly project $\mathcal{F}$ to a similarity scalar as typically done [2, 30, 42]. However, this would ignore the other pose hypotheses, forcing the network to output an absolute score assignment which can be difficult to learn.

Intuitively, we would like the network to leverage the global context of all pose hypotheses in order to make a more informed decision. Therefore, we introduce the second level of comparison among all the K pose hypotheses, or more precisely, the concatenated feature embedding

$\mathbf{F} = [\mathcal{F}_0, \dots, \mathcal{F}_{K-1}]^\top \in \mathbb{R}^{K \times 512}$, which has encoded the pose alignment information. To adapt to varying K, we treat $\mathbf{F}$ as a sequence and perform multi-head self-attention which naturally generalizes to varying lengths [54]. We also do not apply position encoding to $\mathbf{F}$ here, so as to be agnostic to the permutation. The attended feature is then linearly projected to the scores $\mathbf{S} \in \mathbb{R}^K$ to be assigned to each pose hypothesis. The effectivenss of this hierarchical comparison strategy is shown in a typical example in Fig. 4.

Contrast Validation. To train the pose ranking network, we propose a *pose-conditioned triplet loss*:

$$\mathcal{L}(i^+, i^-) = \max(\mathbf{S}(i^-) - \mathbf{S}(i^+) + \alpha, 0), \tag{11}$$

where α denotes the contrastive margin; i^- and i^+ represent the negative and positive pose samples, respectively, which are determined by computing the ADD metric [63] using ground truth. Note that different from standard triplet loss [25], the anchor sample is not shared between the positive and negative samples in our case, since the input is cropped depending on each pose hypothesis to account for translations. While we can compute this loss over each pair in the list, the comparison becomes ambiguous when both poses are far from ground truth. Therefore, we only keep those pose pairs whose positive sample is from a viewpoint that is close enough to the ground truth to make the comparison meaningful:

$$\mathbb{V}^+ = \{i : D(\boldsymbol{R}_i, \bar{\boldsymbol{R}}) < d\} \tag{12}$$

$$\mathbb{V}^- = \{0, 1, 2, \dots, K-1\} \tag{13}$$

$$\mathcal{L}_{\text{rank}} = \sum_{i^+, i^-} \mathcal{L}(i^+, i^-) \tag{14}$$

where the summation is over $i^+ \in \mathbb{V}^+, i^- \in \mathbb{V}^-, i^+ \neq i^-$; $\boldsymbol{R}_i$ and $\bar{\boldsymbol{R}}$ are the rotation of the hypothesis and ground truth, respectively; $D(\cdot)$ denotes the geodesic distance between rotations; and d is a predefined threshold. We also experimented with the InfoNCE loss [44] as used in [42] but observed worse performance (Sec. 4.5). We attribute this to the perfect translation assumption made in [42] which is not the case in our setup.

4. Experiments

4.1. Dataset and Setup

We consider 5 datasets: LINEMOD [21], Occluded LINEMOD [1], YCB-Video [63], T-LESS [23], and YCBInEOAT [59]. These involve various challenging scenarios (dense clutter, multi-instance, static or dynamic scenes, table-top or robotic manipulation), and objects with diverse properties (textureless, shiny, symmetric, varying sizes).

As our framework is unified, we consider the combinations among two setups (model-free and model-based) and two pose prediction tasks (6D pose estimation and tracking), resulting in 4 tasks in total. For the model-free setup, a number of reference images capturing the novel object are selected from the training split of the datasets, equipped with the ground-truth annotation of the object pose, following [20]. For the model-based setup, a CAD model is provided for the novel object. In all evaluation except for ablation, our method always uses the same trained model and configurations for inference *without any fine-tuning*.

	PREDATOR [26]		LoFTR [50]		FS6D-DPM [20]		Ours	
Ref. images	16		16		16		16	
Finetune-free	✓		✓		✗		✓	
Metrics	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD
002_master_chef_can	73.0	17.4	87.2	50.6	92.6	36.8	96.9	91.3
003_cracker_box	41.7	8.3	71.8	25.5	83.9	24.5	97.5	96.2
004_sugar_box	53.7	15.3	63.9	13.4	95.1	43.9	97.5	87.2
005_tomato_soup_can	81.2	44.4	77.1	52.9	93.0	54.2	97.6	93.3
006_mustard_bottle	35.5	5.0	84.5	59.0	97.0	71.1	98.4	97.3
007_tuna_fish_can	78.2	34.2	72.6	55.7	94.5	53.9	97.7	73.7
008_pudding_box	73.5	24.2	86.5	68.1	94.9	79.6	98.5	97.0
009_gelatin_box	81.4	37.5	71.6	45.2	98.3	32.1	98.5	97.3
010_potted_meat_can	62.0	20.9	67.4	45.1	87.6	54.9	96.6	82.3
011_banana	57.7	9.9	24.2	1.6	94.0	69.1	98.1	95.4
019_pitcher_base	83.7	18.1	58.7	22.3	91.1	40.4	97.9	96.6
021_bleach_cleanser	88.3	48.1	36.9	16.7	89.4	44.1	97.4	93.3
024_bowl	73.2	17.4	32.7	1.4	74.7	0.9	94.9	89.7
025_mug	84.8	29.5	47.3	23.6	86.5	39.2	96.2	75.8
035_power_drill	60.6	12.3	18.8	1.3	73.0	19.8	98.0	96.3
036_wood_block	70.5	10.0	49.9	1.4	94.7	27.9	97.4	94.7
037_scissors	75.5	25.0	32.3	14.6	74.2	27.7	97.8	95.5
040_large_marker	81.8	38.9	20.7	8.4	97.4	74.2	98.6	96.5
051_large_clamp	83.0	34.4	24.1	11.2	82.7	34.7	96.9	92.7
052_extra_large_clamp	72.9	24.1	15.0	1.8	65.7	10.1	97.6	94.1
061_foam_brick	79.2	35.5	59.4	31.4	95.7	45.8	98.1	93.4
MEAN	71.0	24.3	52.5	26.2	88.4	42.1	**97.4**	**91.5**

Table 1. Model-free pose estimation results measured by AUC of ADD and ADD-S on YCB-Video dataset. "Finetuned" means the method was fine-tuned with group split of object instances on the testing dataset, as introduced by [20].

4.2. Metric

To closely follow the baseline protocols on each setup, we consider the following metrics:

- Area under the curve (AUC) of ADD and ADD-S [63].
- Recall of ADD that is less than 0.1 of the object diameter (ADD-0.1d), as used in [17, 20].
- Average recall (AR) of VSD, MSSD and MSPD metrics introduced in the BOP challenge [24].

4.3. Pose Estimation Comparison

Model-free. Table 1 presents the comparison results against the state-of-art RGBD methods [20, 26, 50] on YCB-Video dataset. The baselines results are adopted from [20]. Following [20], all methods are given the perturbed ground-truth bounding box as 2D detection for fair comparison. Table 2 presents the comparison results on LINEMOD dataset. The baseline results are adopted from [17, 20]. RGB-based methods [17, 37, 51] are given the privilege of much larger number of reference images to compensate for the lack of depth. Among RGBD methods, FS6D [20] requires fine-tuning on the target dataset. Our method significantly outperforms the existing methods on both datasets without fine-tuning on the target dataset or ICP refinement.

Fig. 5 visualizes the qualitative comparison. We do not have access to the pose predictions of FS6D [20] for qualitative results, since its code is not publicly released. The

Method	Modality	Finetune-free	Ref. images	Objects: ape	benchwise	cam	can	cat	driller	duck	eggbox	glue	holepuncher	iron	lamp	phone	Avg.
Gen6D [37]	RGB	✗	200	-	77	66.1	-	60.7	67.4	40.5	95.7	87.2	-	-	-	-	-
Gen6D* [37]	RGB	✓	200	-	62.1	45.6	-	40.9	48.8	16.2	-	-	-	-	-	-	-
OnePose [51]	RGB	✓	200	11.8	92.6	88.1	77.2	47.9	74.5	34.2	71.3	37.5	54.9	89.2	87.6	60.6	63.6
OnePose++ [17]	RGB	✓	200	31.2	97.3	88.0	89.8	70.4	92.5	42.3	99.7	48.0	69.7	97.4	97.8	76.0	76.9
LatentFusion [46]	RGBD	✓	16	88.0	92.4	74.4	88.8	94.5	91.7	68.1	96.3	94.9	82.1	74.6	94.7	91.5	87.1
FS6D [20]	RGBD	✗	16	74.0	86.0	88.5	86.0	98.5	81.0	68.5	**100.0**	99.5	97.0	92.5	85.0	99.0	88.9
FS6D [20] + ICP	RGBD	✗	16	78.0	88.5	91.0	89.5	97.5	92.0	75.5	99.5	99.5	96.0	87.5	97.0	97.5	91.5
Ours	RGBD	✓	16	**99.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**99.4**	**100.0**	**100.0**	**99.9**	**100.0**	**100.0**	**100.0**	**99.9**

Table 2. Model-free pose estimation results measured by ADD-0.1d on LINEMOD dataset. Gen6D* [37] represents the variation without fine-tuning.

Figure 5. Qualitative comparison of pose estimation on LINEMOD dataset under the model-free setup. Images are cropped and zoomed-in for better visualization.

severe self-occlusion and lack of texture on the glue largely challenge OnePose++ [17] and LatentFusion [46], while our method successfully estimates the pose.

Method	Unseen objects	Dataset: LM-O	T-LESS	YCB-V	Mean
SurfEmb [14] + ICP	✗	75.8	82.8	80.6	79.7
OSOP [48] + ICP	✓	48.2	-	57.2	-
(PPF, Sift) + Zephyr [43]	✓	59.8	-	51.6	-
MegaPose-RGBD [30]	✓	58.3	54.3	63.3	58.6
OVE6D [2]	✓	49.6	52.3	-	-
GCPose [66]	✓	65.2	67.9	-	-
Ours	✓	**78.8**	**83.0**	**88.0**	**83.3**

Table 3. Model-based pose estimation results measured by AR score on representative BOP datasets. All methods use the RGBD modality.

Model-based. Table 3 presents the comparison results among RGBD methods on 3 core datasets from BOP: Occluded-LINEMOD [1], YCB-Video [63] and T-LESS [23]. All methods use Mask R-CNN [16] for 2D detection. Our method outperforms the existing model-based methods that deal with novel objects by a large margin, including the instance-level method [14].

4.4. Pose Tracking Comparison

		se(3)-TrackNet [59]	RGF [27]	Bundle-Track [58]	Bundle-SDF [61]	Wüthrich [62]	Ours	Ours†
Properties	Novel object	✗	✓	✓	✓	✓	✓	✓
	Initial pose	GT	GT	GT	GT	GT	GT	Est.
cracker_box	ADD-S	94.06	55.44	89.41	90.63	88.13	**95.10**	**94.92**
	ADD	90.76	34.78	85.07	85.37	79.00	**91.32**	**91.54**
bleach_cleanser	ADD-S	94.44	45.03	94.72	94.28	68.96	**95.96**	**96.36**
	ADD	89.58	29.40	89.34	87.46	61.47	**91.45**	**92.63**
sugar_box	ADD-S	94.80	16.87	90.22	93.81	92.75	**96.67**	**96.61**
	ADD	92.43	15.82	85.56	88.62	86.78	**94.14**	**93.96**
tomato_soup_can	ADD-S	96.95	26.44	95.13	95.24	93.17	**96.58**	**96.54**
	ADD	93.40	15.13	86.00	83.10	63.71	**91.71**	**91.85**
mustard_bottle	ADD-S	97.92	60.17	95.35	95.75	95.31	**97.89**	**97.77**
	ADD	97.00	56.49	92.26	89.87	91.31	**96.34**	**95.95**
All	ADD-S	95.53	39.90	92.53	93.77	89.18	**96.42**	**96.40**
	ADD	92.66	29.98	87.34	86.95	78.28	**93.09**	**93.22**

Table 4. Pose tracking results measured by AUC of ADD and ADD-S on YCBInEOAT dataset. Ours† represents our unified pipeline that uses the pose estimation module for pose initialization.

Unless otherwise specified, no re-initialization is applied to the evaluated methods in the case of tracking lost, in order to evaluate long-term tracking robustness. We defer to our supplemental materials for qualitative results.

Table 5 presents the comparison results of pose tracking on YCB-Video [63] dataset. Among the baselines, DeepIM [33], se(3)-TrackNet [59] and PoseRBPF [8] need training on the same object instances, while Wüthrich *et al.* [62], RGF [27], ICG [49] and our method can be instantly applied to novel objects when provided with a CAD model.

Solely evaluating on table-top static scenes does not expose challenges of abrupt out-of-plane rotations, dynamic external occlusions and disentangled camera motions [59]. Thus, for more comprehensive comparison, we also evaluate pose tracking methods on the YCBInEOAT [59] dataset which includes videos of dynamic robotic manipulation. Results under the model-based setup are presented in Table 4. Our method achieves the best performance and even outperforms the instance-wise training method [59] with ground-truth pose initialization. Moreover, our unified framework also allows for end-to-end pose estimation and tracking without external pose initialization, which is the only method with such capability, noted as *Ours†* in the table.

4.5. Analysis

Ablation Study. Table 6 presents the ablation study of critical design choices. The results are evaluated by AUC of ADD and ADD-S metrics on the YCB-Video dataset. *Ours (proposed)* is the default version under the model-free (16 reference images) setup. *W/o LLM texture augmentation* removes the LLM-aided texture augmentation for synthetic training. In *W/o transformer*, we replace the transformer-based architecture by convolutional and linear layers while keeping the similar number of parameters. *W/o hierarchical comparison* only compares the rendering and the cropped input trained by pose-conditioned triplet loss (Eq. 11) without two-level hierarchical comparison. At test time, it compares each pose hypothesis with the input observation independently and outputs the pose with the highest score. Example qualitative result is shown in Fig. 4. *Ours-InfoNCE* replaces contrast validated pair-wise loss (Eq. 14) by the InfoNCE loss as used in [42].

Effects of number of reference images. We study how the number of reference images affects the results measured

Approach	DeepIM [33]		se(3)-TrackNet [59]		PoseRBPF [8] + SDF		Wüthrich [62]		RGF [27]		ICG [49]		Ours		Ours†	
Initial pose	GT		GT		PoseCNN		GT		GT		GT		GT		GT	
Re-initialization	Yes (290)		No		Yes (2)		No		No		No		No		No	
Novel object	✗		✗		✗		✓		✓		✓		✓		✓	
Object setup	Model-based		Model-based		Model-based		Model-based		Model-based		Model-based		Model-based		Model-free	
Metric	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S
002_master_chef_can	89.0	93.8	93.9	96.3	89.3	96.7	55.6	90.7	46.2	90.2	66.4	89.7	93.6	97.0	91.2	96.9
003_cracker_box	88.5	93.0	96.5	97.2	96.0	97.1	96.4	97.2	57.0	72.3	82.4	92.1	96.9	97.8	96.2	97.5
004_sugar_box	94.3	96.3	97.6	98.1	94.0	96.4	97.1	97.9	50.4	72.7	96.1	98.4	96.9	98.2	94.5	97.4
005_tomato_soup_can	89.1	93.2	95.0	97.2	87.2	95.2	64.7	89.5	72.4	91.6	73.2	97.3	96.3	98.1	94.3	97.9
006_mustard_bottle	92.0	95.1	95.8	97.4	98.3	98.5	97.1	98.0	87.7	98.2	96.2	98.4	97.3	98.4	97.3	98.5
007_tuna_fish_can	92.0	96.4	86.5	91.1	86.8	93.6	69.1	93.3	28.7	52.9	73.2	95.8	96.9	98.5	84.0	97.8
008_pudding_box	80.1	88.3	97.9	98.4	60.9	87.1	96.8	97.9	12.7	18.0	73.8	88.9	97.8	98.5	96.9	98.5
009_gelatin_box	92.0	94.4	97.8	98.4	98.2	98.6	97.5	98.4	49.1	70.7	97.2	98.8	97.7	98.5	97.6	98.5
010_potted_meat_can	78.0	88.9	77.8	84.2	76.4	83.5	83.7	86.7	44.1	45.6	93.3	97.3	95.1	97.7	94.8	97.5
011_banana	81.0	90.5	94.9	97.2	92.8	97.7	86.3	96.1	93.3	97.7	95.6	98.4	96.4	98.4	95.6	98.1
019_pitcher_base	90.4	94.7	96.8	97.5	97.7	98.1	97.3	97.7	97.9	98.2	97.0	98.8	96.7	98.0	96.8	98.0
021_bleach_cleanser	81.7	90.5	95.9	97.2	95.9	97.0	95.2	97.2	95.9	97.3	92.6	97.5	95.5	97.8	94.7	97.5
024_bowl	38.8	90.6	80.9	94.5	34.0	93.0	30.4	97.2	24.2	82.4	74.4	98.4	95.2	97.6	90.5	95.3
025_mug	83.2	92.0	91.5	96.9	86.9	96.7	83.2	93.3	60.0	71.2	95.6	98.5	95.6	97.9	91.5	96.1
035_power_drill	85.4	92.3	96.4	97.4	97.8	98.2	97.1	97.8	97.9	98.3	96.7	98.5	96.9	98.2	96.3	97.9
036_wood_block	44.3	75.4	95.2	96.7	37.8	93.6	95.5	96.9	45.7	62.5	93.5	97.2	93.2	97.0	92.9	97.0
037_scissors	70.3	84.5	95.7	97s	72.7	85.5	4.2	16.2	20.9	38.6	93.5	97.3	94.8	97.5	95.5	97.8
040_large_marker	80.4	91.2	92.2	96.0	89.2	97.3	35.6	53.0	12.2	18.9	88.5	97.8	96.9	98.6	96.6	98.6
051_large_clamp	73.9	84.1	94.7	96.9	90.1	95.5	61.2	72.3	62.8	80.1	91.8	96.9	93.6	97.3	92.5	96.7
052_extra_large_clamp	49.3	90.3	91.7	95.8	84.4	94.1	93.7	96.6	67.5	69.7	85.9	94.3	94.4	97.5	93.4	97.3
061_foam_brick	91.6	95.5	93.7	96.7	96.1	98.3	96.8	98.1	70.0	86.5	96.2	98.5	97.9	98.6	96.8	98.3
All Frames	82.3	91.9	93.0	95.7	87.5	95.2	78.0	90.2	59.2	74.3	86.4	96.5	**96.0**	**97.9**	**93.7**	**97.5**

Table 5. Pose tracking results measured by AUC of ADD and ADD-S on YCB-Video dataset. Ours† represents our method under the model-free setup with reference images.

	ADD	ADD-S
Ours (proposed)	91.52	97.40
W/o LLM texture augmentation	90.83	97.38
W/o transformer	90.77	97.33
W/o hierarchical comparison	89.05	96.67
Ours-InfoNCE	89.39	97.29

Table 6. Ablation study of critical design choices.

by AUC of ADD and ADD-S on YCB-Video dataset, as shown in Fig. 6. Overall, our method is robust to the number of reference images especially on the ADD-S metric, and saturates at 12 images for both metrics. Notably, even when only 4 reference images are provided, our method still yields stronger performance than FS6D [20] equipped with 16 reference images (Table 1).

Training data scaling law. Theoretically, an unbounded amount of synthetic data can be produced for training. Fig. 7 presents how the amount of training data affects the results measured by AUC of ADD and ADD-S metrics on YCB-Video dataset. The gain saturates around 1M.

Running time. We measure the running time on the hardware of Intel i9-10980XE CPU and NVIDIA RTX 3090 GPU. The pose estimation takes about 1.3 s for one object, where pose initialization takes 4 ms, refinement takes 0.88 s, pose selection takes 0.42 s. Tracking runs much faster at ~32 Hz, since only pose refinement is needed and there are not multiple pose hypotheses. In practice, we can run pose estimation once for initialization and switch to tracking mode for real-time performance.

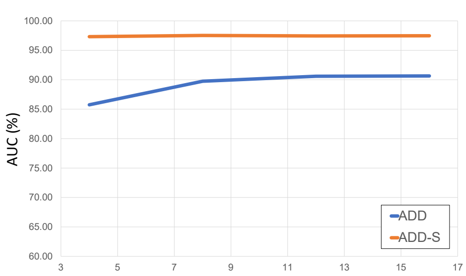

Figure 6. Effects of number of reference images.

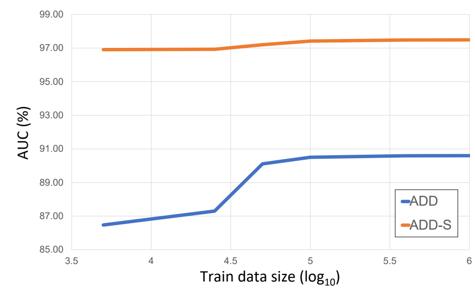

Figure 7. Effects of training data size.

5. Conclusion

We present a unified foundation model for 6D pose estimation and tracking of novel objects, supporting both model-based and model-free setups. Extensive experiments on the combinations of 4 different tasks indicate it is not only versatile but also outperforms existing state-of-art methods specially designed for each task by a considerable margin. It even achieves comparable results to those methods requiring instance-level training. In future work, exploring state estimation beyond single rigid object will be of interest.

References

[1] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D object pose estimation using 3d object coordinates. In *13th European Conference on Computer Vision (ECCV)*, pages 536–551, 2014. 6, 7

[2] Dingding Cai, Janne Heikkilä, and Esa Rahtu. OVE6D: Object viewpoint encoding for depth-based 6D object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6803–6813, 2022. 5, 7

[3] Ming Cai and Ian Reid. Reconstruct locally, localize globally: A model free method for object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3153–3163, 2020. 2

[4] Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and Kangxue Yin. TexFusion: Synthesizing 3D textures with text-guided image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4169–4181, 2023. 2, 3

[5] Dengsheng Chen, Jun Li, Zheng Wang, and Kai Xu. Learning canonical shape space for category-level 6D object pose and size estimation. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 11973–11982, 2020. 1, 2

[6] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3D objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13142–13153, 2023. 2

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 3

[8] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. PoseRBPF: A Rao-Blackwellized particle filter for 6D object pose tracking. In *Robotics: Science and Systems (RSS)*, 2019. 1, 2, 7, 8

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 5

[10] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3D scanned household items. In *International Conference on Robotics and Automation (ICRA)*, pages 2553–2560, 2022. 2

[11] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning (ICML)*, pages 3789–3799, 2020. 4

[12] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5356–5364, 2019. 2

[13] John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, 1996. 5

[14] Rasmus Laurvig Haugaard and Anders Glent Buch. Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6749–6758, 2022. 7

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2961–2969, 2017. 7

[17] Xingyi He, Jiaming Sun, Yuang Wang, Di Huang, Hujun Bao, and Xiaowei Zhou. OnePose++: Keypoint-free one-shot object pose estimation without CAD models. *Advances in Neural Information Processing Systems (NeurIPS)*, 35: 35103–35115, 2022. 1, 2, 6, 7

[18] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. PVN3D: A deep point-wise 3D keypoints voting network for 6DoF pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11632–11641, 2020. 1, 2

[19] Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. FFB6D: A full flow bidirectional fusion network for 6D pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3003–3013, 2021. 1, 2

[20] Yisheng He, Yao Wang, Haoqiang Fan, Jian Sun, and Qifeng Chen. FS6D: Few-shot 6D pose estimation of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6814–6824, 2022. 2, 3, 6, 7, 8

[21] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *International Conference on Computer Vision (ICCV)*, pages 858–865, 2011. 6

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:6840–6851, 2020. 2

[23] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888, 2017. 6, 7

[24] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. BOP: Benchmark

for 6D object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018. 2, 6

[25] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *Third International Workshop on Similarity-Based Pattern Recognition (SIMBAD)*, pages 84–92, 2015. 6

[26] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. PREDATOR: Registration of 3D point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4267–4276, 2021. 6

[27] Jan Issac, Manuel Wüthrich, Cristina Garcia Cifuentes, Jeannette Bohg, Sebastian Trimpe, and Stefan Schaal. Depth-based object tracking using a robust gaussian filter. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 608–615, 2016. 1, 2, 7, 8

[28] Daniel Kappler, Franziska Meier, Jan Issac, Jim Mainprice, Cristina Garcia Cifuentes, Manuel Wüthrich, Vincent Berenz, Stefan Schaal, Nathan Ratliff, and Jeannette Bohg. Real-time perception meets reactive motion generation. *IEEE Robotics and Automation Letters*, 3(3):1864–1871, 2018. 1

[29] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent multi-view multi-object 6D pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 574–591, 2020. 1, 2

[30] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. MegaPose: 6D pose estimation of novel objects via render & compare. In *6th Annual Conference on Robot Learning (CoRL)*, 2022. 1, 2, 5, 7

[31] Taeyeop Lee, Jonathan Tremblay, Valts Blukis, Bowen Wen, Byeong-Uk Lee, Inkyu Shin, Stan Birchfield, In So Kweon, and Kuk-Jin Yoon. TTA-COPE: Test-time adaptation for category-level object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21285–21295, 2023. 1, 2

[32] Fu Li, Shishir Reddy Vutukur, Hao Yu, Ivan Shugurov, Benjamin Busam, Shaowu Yang, and Slobodan Ilic. NeRF-Pose: A first-reconstruct-then-regress approach for weakly-supervised 6D object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2123–2133, 2023. 2

[33] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep iterative matching for 6D pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018. 1, 2, 7, 8

[34] Zhigang Li, Gu Wang, and Xiangyang Ji. CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation. In *CVF International Conference on Computer Vision (ICCV)*, pages 7677–7686, 2019. 2

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *13th European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 3

[36] Yunzhi Lin, Jonathan Tremblay, Stephen Tyree, Patricio A Vela, and Stan Birchfield. Keypoint-based category-level object pose tracking from an RGB sequence with uncertainty estimation. In *International Conference on Robotics and Automation (ICRA)*, 2022. 1, 2

[37] Yuan Liu, Yilin Wen, Sida Peng, Cheng Lin, Xiaoxiao Long, Taku Komura, and Wenping Wang. Gen6D: Generalizable model-free 6-DoF object pose estimation from RGB images. *ECCV*, 2022. 1, 2, 6, 7

[38] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 4

[39] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: A hands-on survey. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 22(12):2633–2651, 2015. 1

[40] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 4

[41] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 4

[42] Van Nguyen Nguyen, Yinlin Hu, Yang Xiao, Mathieu Salzmann, and Vincent Lepetit. Templates for 3D object pose estimation revisited: Generalization to new objects and robustness to occlusions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6771–6780, 2022. 5, 6, 7

[43] Brian Okorn, Qiao Gu, Martial Hebert, and David Held. Zephyr: Zero-shot pose hypothesis rating. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 14141–14148, 2021. 7

[44] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 6

[45] Kiru Park, Timothy Patten, and Markus Vincze. Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7668–7677, 2019. 1, 2

[46] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. LatentFusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10710–10719, 2020. 2, 7

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 2

[48] Ivan Shugurov, Fu Li, Benjamin Busam, and Slobodan Ilic. OSOP: A multi-stage one shot object pose estimation framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6835–6844, 2022. 1, 2, 7

[49] Manuel Stoiber, Martin Sundermeyer, and Rudolph Triebel. Iterative corresponding geometry: Fusing region and depth for highly efficient 3D tracking of textureless objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6855–6865, 2022. 1, 2, 7, 8

[50] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8922–8931, 2021. 6

[51] Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. OnePose: One-shot object pose estimation without CAD models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6825–6834, 2022. 1, 2, 6, 7

[52] Meng Tian, Marcelo H Ang, and Gim Hee Lee. Shape prior deformation for categorical 6D object pose and size estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 530–546, 2020. 1, 2

[53] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. In *Conference on Robot Learning (CoRL)*, pages 306–316, 2018. 2

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017. 5, 6

[55] Chen Wang, Roberto Martín-Martín, Danfei Xu, Jun Lv, Cewu Lu, Li Fei-Fei, Silvio Savarese, and Yuke Zhu. 6-PACK: Category-level 6D pose tracker with anchor-based keypoints. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 10059–10066, 2020. 1, 2

[56] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6D object pose and size estimation. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 2642–2651, 2019. 1, 2

[57] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 4

[58] Bowen Wen and Kostas Bekris. BundleTrack: 6D pose tracking for novel objects without instance or category-level 3D models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8067–8074, 2021. 2, 7

[59] Bowen Wen, Chaitanya Mitash, Baozhang Ren, and Kostas E Bekris. se(3)-TrackNet: Data-driven 6D pose tracking by calibrating image residuals in synthetic domains. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10367–10373, 2020. 1, 2, 6, 7, 8

[60] Bowen Wen, Wenzhao Lian, Kostas Bekris, and Stefan Schaal. CatGrasp: Learning category-level task-relevant grasping in clutter from simulation. In *International Conference on Robotics and Automation (ICRA)*, pages 6401–6408, 2022. 1

[61] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. BundleSDF: Neural 6-DoF tracking and 3D reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 606–617, 2023. 2, 4, 7

[62] Manuel Wüthrich, Peter Pastor, Mrinal Kalakrishnan, Jeannette Bohg, and Stefan Schaal. Probabilistic object tracking using a range camera. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3195–3202, 2013. 1, 2, 7, 8

[63] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*, 2018. 2, 6, 7

[64] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:2492–2502, 2020. 4

[65] Ruida Zhang, Yan Di, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. SSP-Pose: Symmetry-aware shape prior deformation for direct category-level object pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7452–7459, 2022. 1, 2

[66] Heng Zhao, Shenxing Wei, Dahu Shi, Wenming Tan, Zheyang Li, Ye Ren, Xing Wei, Yi Yang, and Shiliang Pu. Learning symmetry-aware geometry correspondences for 6D object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14045–14054, 2023. 7

[67] Linfang Zheng, Chen Wang, Yinghan Sun, Esha Dasgupta, Hua Chen, Aleš Leonardis, Wei Zhang, and Hyung Jin Chang. HS-Pose: Hybrid scope feature extraction for category-level object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17163–17173, 2023. 2

[68] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5745–5753, 2019. 5