Finding NeMO: A Geometry-Aware Representation of Template Views for Few-Shot Perception

Sebastian Jung Leonard Klüpfel Rudolph Triebel Maximilian Durner
German Aerospace Center (DLR)
{Sebastian.Jung, Leonard.Kluepfel, Rudolph.Triebel, Maximilian.Durner}@dlr.de

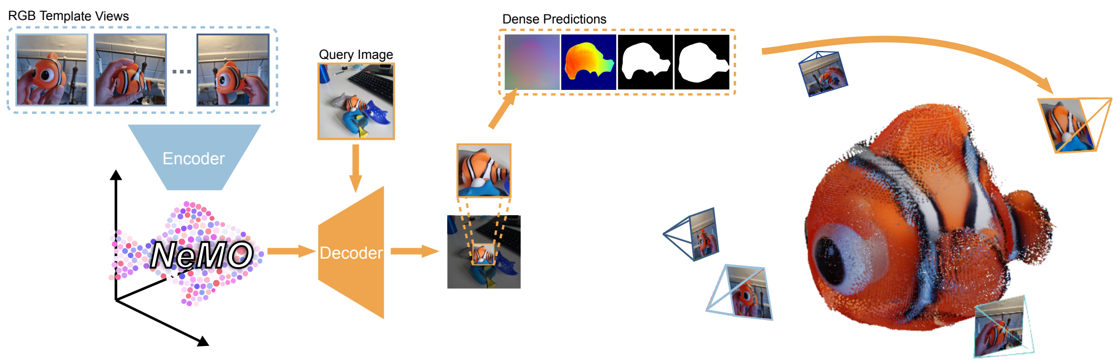

Figure 1. **Overview.** Our method uses a multi-view encoder to generate an object-centric geometric encoding called *Neural Memory Object (NeMO)* with its own coordinate system from a set of RGB images depicting an object unseen during training. A decoder uses the NeMO to retrieve dense predictions allowing us to detect, segment, estimate the objects surface and determine the camera-to-object position on an RGB query image. Even in cluttered scenes, our method is able to find the object, which we can use to crop the corresponding region of interest, demonstrating that our method can be used for multi-stage perception pipelines. Images were captured using a normal smartphone.

Abstract

We present Neural Memory Object (NeMO), a novel object-centric representation that can be used to detect, segment and estimate the 6DoF pose of objects unseen during training using RGB images. Our method consists of an encoder that requires only a few RGB template views depicting an object to generate a sparse object-like point cloud using a learned UDF containing semantic and geometric information. Next, a decoder takes the object encoding together with a query image to generate a variety of dense predictions. Through extensive experiments, we show that our method can be used for few-shot object perception without requiring any camera-specific parameters or retraining on target data. Our proposed concept of outsourcing object information in a NeMO and using a single network for multiple perception tasks enhances interaction with novel objects, improving scalability and efficiency by enabling quick object onboarding without retraining or extensive pre-processing. We report competitive and state-of-the-art results on various datasets and perception tasks of the BOP benchmark, demonstrating the versatility of our approach. https://github.com/DLR-RM/nemo

1. Introduction

Objects play a central role in our daily lives, and recognizing them in images is critical for applications such as robotics, augmented reality, and autonomous systems. Recent advances in deep learning and computer vision have greatly improved object perception, especially in *model-based* approaches that leverage 3D CAD models to train deep networks for detection, segmentation, and pose estimation [25]. These methods benefit from large-scale synthetic training [11] and can specialize in specific objects or categories, achieving impressive performance when 3D models are available at test time. However, in many real-world scenarios, it is impractical to assume access to a 3D model for every object. As a result, *model-free perception* has become a growing research focus, aiming to rapidly onboard and recognize novel objects.

When a CAD model is unavailable during training but provided at inference, recent methods adapt perception models using rendered templates of the target object, leveraging the geometric and textural cues from these synthetic views. Some approaches fine-tune object-specific networks in minimal time [58], while others use general-purpose features to recognize new objects without retraining [43]. A common strategy compares features between template views and query images, followed by post-processing [29, 44]. However, these approaches scale poorly with the number of templates and rely on pairwise local comparisons [12, 50, 53], without jointly reasoning over all views. While still affected by the sim-to-real gap, such methods enable fast onboarding of CAD models.

In model-free object perception – where no CAD model is available at any stage – the challenge is greater. Synthetic training tailored to the object is infeasible, and geometric information must be extracted from real reference images, where object-to-camera poses are typically unknown. Some methods rely on template or local feature matching using real images [22, 32], while others employ neural fields [38] to reconstruct CAD-like geometry [61, 62], requiring extrinsic pose information.

To overcome the limitations of existing perception systems in terms of generalization, scalability, and efficiency, we propose a novel encoder-decoder architecture trained on a large-scale synthetic dataset. Our method constructs a geometry-aware representation, termed Neural Memory Object (NeMO), from a set of unordered, object-centric RGB images, without requiring camera calibration or pose annotations. NeMO is formulated as a sparse, continuous point cloud that encapsulates both semantic and geometric features observed from multiple viewpoints. Unlike conventional encoder-decoder models that compress inputs into a single latent vector [51, 55], NeMO preserves a structured, interpretable abstraction of object geometry, enabling transformations such as translation, rotation, and scaling without reliance on a 3D CAD model. Critically, the object-specific information is disentangled from the decoder's parameters, enabling object-agnostic inference and robust generalization to novel instances not seen during training. The point-based representation supports incremental refinement through the addition of new views, without necessitating reprocessing of previous observations. Furthermore, the decoupling of encoding and decoding facilitates efficient deployment, as NeMO can be precomputed offline, rendering inference time invariant to the number of input views and improving scalability for real-world applications.

We extensively ablate the representation to analyze its potential and quantitatively demonstrate its performance on multiple object perception tasks, achieving competitive and state-of-the-art results on model-free and model-based unseen object benchmarks. Additionally, we qualitatively show its potential for object surface reconstruction. Concretely, our contributions are threefold:

- We propose the **NeMO**, a geometry-sensitive, and compact representation of template views well-suited for few-shot object perception tasks.
- We evaluate our encoder-decoder network on the task of model-based and -free **few-shot detection, segmentation and pose estimation of unseen objects**. Without training on test objects, we perform competitively and partially better against state-of-the-art methods.
- We contribute an object-centric **diverse synthetic dataset**, mimicking realistic cluttered scenes that balances the occurrences of different object classes to advance research on related topics.

2. Related Work

Our work focuses on the perception of previously *unseen* objects, encompassing detection, segmentation, and pose estimation. *Unseen* refers to objects not explicitly seen during training, provided at runtime via 3D models (*model-based*) or a set of RGB template images (*model-free*). While designed for the model-free setting, our method also supports model-based input.

Unseen Object Segmentation and Detection. Given a target object, GFreeDet [35], a model-free method, detects and segments objects through reconstructing a Gaussian object and comparing generated templates to region proposal from the query image. NOCTIS [17] similarly computes segmentation masks by matching the appearance and semantic of reference images and a query image as obtained through Grounded SAM2 and DINOv2.

Instead of learning a presentation to render template images, model-based approaches directly use the provided CAD-model. To this end, CNOS [43] matches DINOv2 [45] tokens of template renderings with region proposals derived from SAM/ FastSAM [28, 66]. Similarly, NIDS-Net [37] also leverages SAM [28] and Grounding DINO [34] to obtain proposal embeddings which are compared to averaged and masked DINOv2's template embeddings of the object. OC-DiT [57] uses latent diffusion models conditioned on template images to generate segmentation masks.

Unseen Object Pose Estimation. Model-free pose estimation also commonly requires multiple reference registration images [31, 47, 54, 62]. To this end, FS6D [22] predicts 6D poses by fusing features from support RGBD images and the query scene to establish dense correspondences. OnePose [54] and OnePose++ [21] employ pairwise image matching of dense 2D-3D correspondences hierarchically in a coarse-to-fine fashion. RelPose [65] estimates the relative camera rotation between image pairs through an energy-based formulation, which is extended to the relative 6D pose based on multiple images in the extension RelPose++ [32]. In contrast, PIZZA [42] approaches 6D object tracking through either an image pair or multiple template images between which the relative pose is estimated. Similarly, BundleSDF [61] jointly tracks poses and learns a Neural Object Field from a RGB-D stream, using RGB renderings from the learned field to supervise texture and geometry prediction. FoundationPose [62] estimates ei-

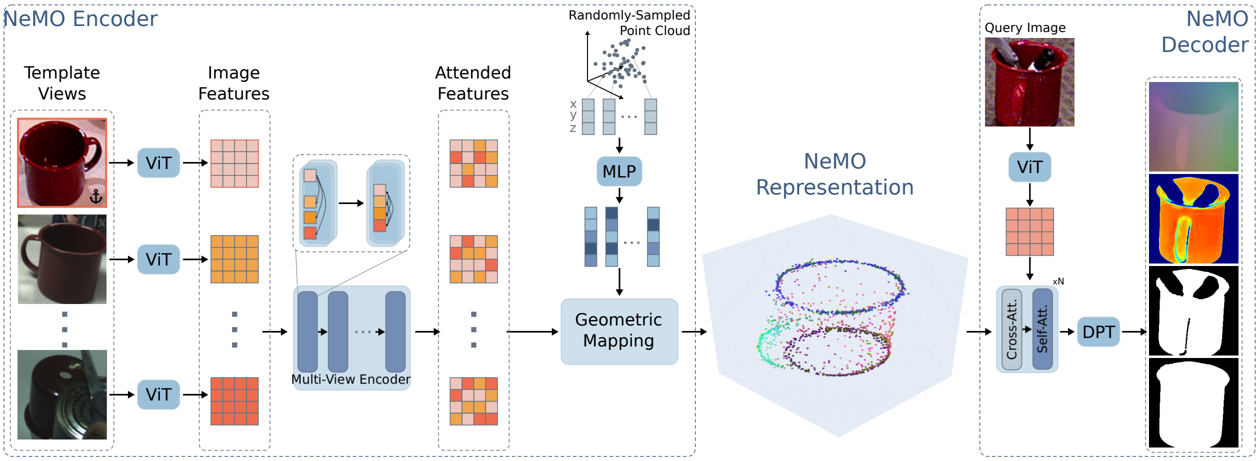

Figure 2. **Overview of the NeMO approach.** RGB template views are first processed by a ViT [13] and a multi-view encoder [27], producing updated image features. To incorporate spatial information, a randomly-sampled point cloud is processed through a MLP and attended to the image features in our proposed *Geometric Mapping* block, yielding feature-enhanced 3D points that form the NeMO. A decoder attends a query image with the NeMO using multiple Cross- and Self-Attention blocks to generate multiple dense predictions. We represent the NeMO as a point cloud and reduce the higher dimensional NeMO features to RGB using PCA [49]. The PCA reduction shows a relation between the learned features and the objects geometry and semantics.

ther single-view poses for a model-based or a model-free scenario and also performs pose tracking. The model-free setup is similar to BundleSDF, whereas in the model-based scenario templates are rendered based on the given model.

Analogously to detection and segmentation, model-based pose estimation relies on a CAD model of the unseen target object. GigaPose [44] relies on comparing and matching the most similar RGB template to the query image. Mega-Pose [29] applies a render-and-compare strategy to refine the best reference rendering of the target object. Both methods rely on pre-training on a large-scale synthetic dataset spanning 2 million images. OSOP [52] establishes dense 2D-3D correspondences based on a template match, similar to ZS6D [1], which also relies on a self-supervised pre-trained ViT for feature extraction. SAM6D [33] predicts the 6D pose and the segmentation mask by applying a coarse-to-fine 3D-3D correspondence matching strategy that builds up on region proposal obtained from SAM and matched with template renderings w.r.t. appearance, semantics and geometry. Similarly, ZeroPose [9] also estimates the 6D pose and the segmentation masks given the CAD model and an RGB-D image as input. The method matchs the query image with the feature embeddings derived from DINOv2 and prompted with the CAD model.

3. Method

In the following, we present our encoder-decoder network as well as the proposed Neural Memory Object (NeMO) representation. Figure 2 gives an overview of the complete approach. Our core idea is to separate visual and geometric object information from neural network weights, enabling an object-agnostic approach that adapts to any demonstrated target object without additional training. To this end, given a set of RGB template images $\mathcal{I} = \{I_i\}_{i=1}^{K}$ of an object, where $K \geq 2$ and $I_i \in \mathbb{R}^{H \times W \times 3}$ with H and W being the image height and width, we aim to construct a unified, object-centric representation without the need for intrinsic or extrinsic camera parameters.

3.1. Network Architecture

NeMO Encoder. As Leap [27] has shown strong generalizability to novel objects, we use a similar attention mechanism in our network. A Vision Transformer (ViT) [13] is used to extract patch-wise image features from all template images $\mathcal{F}^{\mathcal{I}} = \{f_i^{\mathcal{I}}\}_{i=1}^{N}$ with $f_i^{\mathcal{I}} \in \mathbb{R}^d$ and $N = H_{\text{patch}} \times W_{\text{patch}} \times K$ where $H_{\text{patch}}, W_{\text{patch}}$ are the number of vertically and horizontally extracted patches respectively. Since no prior information about the object coordinate frame or the camera-to-object transformation is provided, we define an anchor image I_{A}. It serves as the object's initial orientation in the NeMO space. During training, a random image from $\mathcal{I}$ acts as anchor I_{A}. As in [27], we employ a multi-view encoder to generate updated image features $\widehat{\mathcal{F}}^{\mathcal{I}}$. This part incorporates a strong bias towards the anchor image, using cross- and self-attention blocks that facilitate interactions between the anchor and non-anchor features.

Geometric Mapping. To enhance these features with the 3D geometry of the object oriented in the anchor coordinate system, we first create $\mathcal{Q} = \{q_i\}_{i=1}^{M}$, a set of sampled 3D points $q_i \in [-1, 1]^3$ with $M \geq 1$. Next, we train a Multi-Layer Perceptron (MLP) (see Supplementary Fig. 7) to act as point encoder λ that maps each 3D point q_i to a corresponding feature vector $\mathcal{F}^{\mathcal{Q}} = \{f_i^{\mathcal{Q}}\}_{i=1}^{M} = \{\lambda(q_i)\}_{i=1}^{M}$ with $f_i^{\mathcal{Q}} \in \mathbb{R}^d$. We fuse the resulting set of point features $\mathcal{F}^{\mathcal{Q}}$ and the updated image features $\widehat{\mathcal{F}}^{\mathcal{I}}$ via our proposed *geometric mapping block*. As shown in Fig. 3, the initial point features $\mathcal{F}^{\mathcal{Q}}$ are updated with the information from $\widehat{\mathcal{F}}^{\mathcal{I}}$ through multiple Transformer Decoders:

$$\widehat{\mathcal{F}}^{\mathcal{Q}} = \left\{\widehat{f}_i^{\mathcal{Q}}\right\}_{i=1}^{M} = \mathrm{TransformerDec}(\mathcal{F}^{\mathcal{Q}}, \widehat{\mathcal{F}}^{\mathcal{I}}) \in \mathbb{R}^d, \tag{1}$$

where $\mathcal{F}^{\mathcal{Q}}$ acts as queries and $\widehat{\mathcal{F}}^{\mathcal{I}}$ as key-value pairs. This allows the initial 3D point cloud features to attend to object-specific 2D image features. Besides the visual cues, we also want to encode shape information of the object. Therefore, we jointly learn a Unsigned Distance Field (UDF) $\mathcal{U}$ using an MLP that predicts the unsigned distance from each point q_i in the initial point cloud $\mathcal{Q}$ to its closest point $s_i \in [-1, 1]^3$ on the estimated object surface $\mathcal{S} = \{s_i\}_{i=1}^{M}$ w.r.t. the coordinate system of I_{A}. Note that $|\mathcal{Q}| = |\mathcal{S}|$. Since $\widehat{f}_i^{\mathcal{Q}}$ depends on q_i we can define the distance d_i and direction v_i as

$$d_i = \mathcal{U}\left(\widehat{f}_i^{\mathcal{Q}}(q_i)\right) \in \mathbb{R} \quad \text{and} \quad v_i = \frac{d\mathcal{U}\left(\widehat{f}_i^{\mathcal{Q}}(q_i)\right)}{dq_i} \in \mathbb{R}^3, \tag{2}$$

such that $s_i = q_i - d_i v_i$. Note that d_i is a scalar.

NeMO Representation. Given the M estimated surface points from the set $\mathcal{S}$ and the processed 3D point features $\widehat{\mathcal{F}}^{\mathcal{Q}}$, we define our NeMO representation as $\chi = \left\{\left(s_i, \widehat{f}_i^{\mathcal{Q}}\right)\right\}_{i=1}^{M}$. Keeping s_i and $\widehat{f}_i^{\mathcal{Q}}$ separate, we can transform the point cloud in NeMO space, allowing us to free χ from the anchor coordinate system defined by I_{A}. This gives the advantage of being able to modify the geometric prediction of downstream tasks. For convenience, we define the complete NeMO encoder (see Fig. 2) as a neural network Ψ such that $\chi = \Psi(\mathcal{I}, \mathcal{Q})$. To fuse the information of the predicted 3D points s_i and their corresponding features we set $\widetilde{f}_i^{\mathcal{Q}} = \widehat{f}_i^{\mathcal{Q}} + \lambda(s_i)$. Our continuous UDF approach differs from Leap [27], which relies on a discrete, fixed-size neural volume. We design our architecture to allow (i) variation in the number of points M, adjusting the size and descriptiveness of χ, (ii) biased point sampling for $\mathcal{Q}$, such as surface points from a CAD model, (iii) transformations of the NeMO point cloud, allowing object modifications after the NeMO generation, and (iv) extension of an existing NeMO with another set of NeMO points.

NeMO Decoder. When observing a query image $I_{\mathrm{q}} \notin \mathcal{I}$ the goal of our decoder Θ is to use the information stored in a NeMO, which integrates information from all images in $\mathcal{I}$, to return a set of perception related dense predictions for the query image. As a first step the query image features $\mathcal{F}^{\mathrm{q}} = \{f_i^{\mathrm{q}}\}_{i=1}^{H_{\text{patch}} \times W_{\text{patch}}}$ obtained by a ViT are updated by

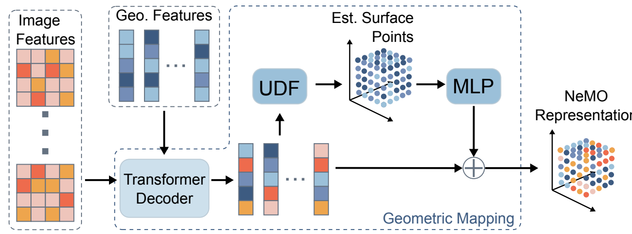

Figure 3. **Geometric Mapping Block.** We fuse the updated image features (key-value pairs) with the pre-processed geometric features (queries) in multiple transformer decoder blocks. The features are then forwarded to a UDF that estimates the unsigned distance of the initial point cloud to the estimated object surface. After further processing these points via a MLP, we combine them with the updated geometric features, resulting in the NeMO.

the information stored in $\widetilde{f}_i^{\mathcal{Q}}$ using a combination of cross- and self-attention layers as shown in Fig. 2. Inspired by the recent advancements in camera pose estimation and dense predictions [31, 59, 60], we use DPT [48] with multiple output heads to upscale the updated query image features $\widehat{\mathcal{F}^{\mathrm{q}}}$ to multiple dense outputs $\Theta(\chi, I_{\mathrm{q}}) = (P_{\text{modal}}, P_{\text{amodal}}, X, C)$, where $P_{\text{modal}} \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}}$ and $P_{\text{amodal}} \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}}$ are the predicted modal and amodal segmentation masks of the object, $X \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times 3}$ is the predicted dense pointmap between 2D pixels in I_{q} and their corresponding 3D surface points in the coordinate system defined by χ and $C \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}}$ is the associated learned confidence map to assess the 2D-3D mapping accuracy of X. After filtering the estimated pointmap based on C, we utilize RANSAC [16] and Perspective-n-Point (PnP) [20, 30] to estimate the pose of the object in the query image.

3.2. Training

We train the encoder Ψ and decoder Θ jointly end-to-end through multiple losses. During training, we have access to a dataset of synthetically rendered RGB-D images with ground truth object poses, amodal and modal masks and camera intrinsic. Given a training sample of multiple images of the same object, we randomly select one image as the anchor image I_{A} and one as a query image I_{q} and use the ground truth poses together with the masked depth and camera intrinsic to build the ground truth surface points of the initial orientation of the NeMO space.

Losses. To enforce the encoder to predict towards ground truth surface points we define a simple regression loss that minimizes the Euclidean distance between the estimated surface point $s_i = q_i - \widetilde{\mathcal{U}}(q_i) \times \frac{d\mathcal{U}(q_i)}{dq_i}$ and the ground truth surface point $\bar{s}_i$:

$$\mathrm{L}_{\chi} = \frac{\sum_{i=1}^{M} \|s_i - \bar{s}_i\|}{M}, \tag{3}$$

with $M = |\chi|$. We do not directly enforce any loss on the NeMO features $\widehat{\mathcal{F}}^{\mathcal{Q}}$ so that they can be freely learned through backpropagation through the decoder. During training, after Ψ has predicted χ, we randomly rotate, translate and scale the NeMO points s_i by a random transformation $\mathbf{T}$ to teach the decoder to learn a coordinate system that is independent of the anchor image I_A. We define additional losses on the dense predictions of the decoder Θ. For the modal and amodal segmentation losses L_modal and L_amodal we use an equally weighted dice-loss [39] and binary cross-entropy loss [26]. The pointmap loss L_2D3D is a confidence weighted L1 loss which uses the estimated confidence map C to weight the loss between the estimated 2D-3D correspondences X and the ground truth mapping $\bar{X}$. To learn the confidence map C, we define a certainty loss L_certain and an uncertainty loss $\text{L}_\text{uncertain}$:

$$\begin{aligned} \text{L}_\text{certain} &= \frac{\sum\limits_{i \in \mathcal{D}_\text{Obj}} \Big(1 - \tanh\big(\exp(C_i)\big)\Big)}{|\mathcal{D}_\text{Obj}|} \\ \text{L}_\text{uncertain} &= \frac{\sum\limits_{i \in \mathcal{D}_\text{Bg}} \Big(\tanh\big(\exp(C_i)\big)\Big)}{|\mathcal{D}_\text{Bg}|} \end{aligned} \tag{4}$$

where $\mathcal{D}_\text{Obj}$ are the pixels belonging to the object and $\mathcal{D}_\text{Bg}$ are all pixels belonging to the background. The total loss is a weighted sum between all losses. More details about the training in Supplementary Sec. 7.2.

4. Experiments

Synthetic Dataset. We create a new object-centric dataset using BlenderProc [11] to generate Physically-Based Rendering (PBR) images given the CAD object models as provided by a subset of Objaverse [10], GSO [14], and OmniObject3D [63], resulting in a total of 11077 different objects. We deem this necessary as there is – to the best of our knowledge – no available dataset of comparable object variety with sufficiently high but also balanced distribution of views per object. For the following experiments, a single network is trained on parts of Objaverse and all OmniObject3D models. We supervised the training by evaluating on GSO objects, no fine-tuning or training on any of the objects present in the BOP benchmark is performed. For additional information we refer to Supplementary Sec. 7.1.

Implementation. We train our method on the aforementioned synthetic dataset and resize the respective, artificially corrupted object bounding box crops to 224×224. We train for 400k steps (roughly 2000 epochs) with a maximum learning rate of 1×10^{-4} on which we apply a linear warm-up of 5000 steps followed by standard cosine annealing. The ViT backbone is trained with a separate maximum learning rate of 1×10^{-5}. As ViT we use DINOv2 [45] and DPT [48] as regression head. Optimization is performed using AdamW [36]. Training takes roughly 10 days on 16 A100 GPUs with an effective batch size of 128, whereas the following experiments are run on a single A100 GPU.

Method	HOPEv2	HANDAL
CNOS (SAM) - Static onboarding [43]	0.345	–
dounseen-SAM-CTL [18]	0.380	–
GFreeDet-FastSAM [35]	0.364	0.255
GFreeDet-SAM [35]	<u>0.384</u>	<u>0.264</u>
Ours	**0.411**	**0.273**

Table 1. **Model-Free Detection.** We compare AP on BOP test splits of HOPEv2 and HANDAL against other methods published on the public Model-Free Unseen Object 2D Detection leaderboard [5].

Method	Detections	HOPEv2	HANDAL
OPFormer†	CNOS [43]	**0.335**	0.204
Ours	CNOS [43]	0.307	–
Ours	GFreeDet-FastSAM [35]	<u>0.329</u>	<u>0.213</u>
Ours	NeMO	0.302	**0.235**

Table 2. **Model-Free 6DoF Pose Estimation.** We compare AP on BOP test splits of HOPEv2 and HANDAL against other methods published on the public Model-Free Unseen Object 6D Detection leaderboard [6]. † indicates unpublished methods.

Experimental Setup. We evaluate our method's capability to perform multiple few-shot perception tasks in a model-free setting, *i.e.* no CAD model is given, and a model-based setting, *i.e.* a CAD model is given only during inference but not during training. Following the BOP challenge's [25] dataset split we use the T-LESS [24], TUD-L [23] and YCB-V [64] datasets for model-based evaluation and the HOPEv2 [56] and HANDAL [19] datasets for model-free evaluation. As metrics, we use *Average Precision (AP)* and *Average Recall (AR)* as defined in [25]. In the model-free setup we use 32 randomly picked real RGB templates from the static onboarding videos provided by the BOP benchmark to generate the NeMO representation while in the model-based setting 32 PBR rendered images are used. The NeMO coordinate system is aligned with the ground truth object coordinate system as described in Supplementary Sec. 7.4 for evaluation purposes only. The dense decoder outputs are used for detection, segmentation and 6DoF pose estimation as described in Supplementary Sec. 7.5. Note that in both settings, the network weights are not changed, *i.e.* no finetuning is performed and the objects have never been seen during training. Additionally, we extensively ablate the NeMO representation and its influence on the downstream tasks in Sec. 4.3 and show qualitative object surface reconstruction results of unknown objects in Sec. 4.1. The same network is used for all experiments if not stated otherwise.

4.1. Model-Free Few-Shot Perception

Model-Free Detection. Tab. 1 shows the AP results of our model-free amodal detection on HOPEv2 and HAN-

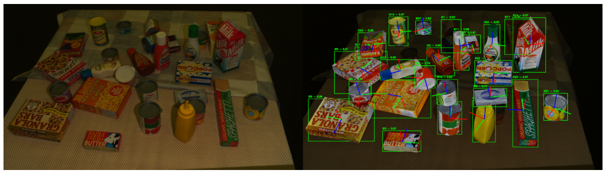

Figure 4. **Qualitative Example of Model-Free Few-Shot Detection and Pose Estimation on HOPEv2.** Left shows the scene without annotations, right shows NeMO detections in green and pose estimations with refinement as rendered overlays. Even in the underexposed scene the model predicts reasonable results.

DAL datasets compared to all other publicly listed results. Our method achieves state-of-the art performance on both datasets, outperforming the previous best method by 2.7pp on HOPEv2 and 1.9pp on HANDAL. While all other methods rely on SAM [28, 66] segmentations of the scene for their bounding box predictions, we are, to the best of our knowledge, the first to use a single network to predict amodal segmentations/detections in a model-free setting. While SAM is able to give modal object segmentations, we can predict amodal segmentations based on the NeMO representation, which we use to create amodal bounding boxes. Note that in the model-free category, the BOP benchmark does only evaluate amodal detection, no segmentation.

Model-Free 6DoF Pose Estimation. We evaluate our model's capability for 6DoF Pose Estimation in a model-free setting on the HOPEv2 and HANDAL datasets using different detections in Tab. 2. On HOPEv2 we use ICP [7] between our estimated object surface and the depth information while no refinement is used on HANDAL, since no depth data is available. An example can be seen in Fig. 4. Compared to the only other method OPFormer we achieve state-of-the-art results on HANDAL when using NeMO detections while being on par when using GFreeDet-FastSAM detections. When using the default CNOS detection as provided by the BOP benchmark, we are 2.8pp behind OPFormer. Note that as of the time of writing, no default CNOS detections for HANDAL are available anymore. Although our detections outperform previous methods on HOPEv2 they do not provide an AP gain for pose estimation. We hypothesize that the detection improvements come from our models ability to predict amodal bounding boxes, which is beneficial for the detection evaluation but might not always be an improvement for pose estimation.

Model-Free Object Reconstruction. We qualitatively show object surface reconstruction on random objects in different scenarios in Fig. 5.

4.2. Model-Based Few-Shot Perception

This section discusses results of our network on model-based perception. Although the network has never been trained on rendered template images with black background and real query images, it performs on par and partially outperforms previous methods specialized on this task.

Method	T-LESS	TUD-L	YCB-V
CNOS(Fast Sam) [43]	0.395	0.534	0.568
SAM6D-FastSAM [33]	0.417	0.546	0.573
SAM6D [33]	0.458	0.573	0.589
F3Dt2D†	0.482	0.573	0.666
MUSE†	0.467	0.590	0.674
anonymity†	0.477	0.593	**0.685**
NIDS Net [37]	**0.493**	0.486	0.621
Ours	0.183	**0.623**	0.602

Table 3. **Model-Based Detection.** We compare AP on BOP test splits of T-LESS, TUD-L and YCB-V against other methods published on the public Model-Based Unseen Object 2D Detection leaderboard [2]. † indicates unpublished methods.

Method	T-LESS	TUD-L	YCB-V
CNOS(Fast Sam) [43]	0.374	0.480	0.599
SAM6D-FastSAM [33]	0.420	0.517	0.621
SAM6D [33]	0.451	0.569	0.605
NOCTIS [17]	0.479	0.583	0.684
LDSeg†	0.488	0.587	0.647
MUSE†	0.451	0.565	0.672
Prisma-MPG + SG†	0.454	**0.590**	0.607
anonymity†	0.464	0.569	**0.688**
NIDS Net [37]	**0.496**	0.556	0.650
Ours	0.169	0.488	0.579

Table 4. **Model-Based Segmentation.** We compare AP on BOP test splits of T-LESS, TUD-L and YCB-V against other methods published on the public Model-Based Unseen Object 2D Segmentation leaderboard [4]. † indicates unpublished methods.

Method	Detections	T-LESS	TUD-L	YCB-V
Ours	NeMO	0.082	0.466	0.493
Ours	ground truth	0.295	0.538	0.566
ZS6D [1]	CNOS [43]	0.210	–	0.324
MegaPose [29]	CNOS [43]	0.177	0.258	0.281
GenFlow [40]	CNOS [43]	0.215	0.300	0.277
GigaPose [44]	CNOS [43]	0.264	0.300	0.278
FoundPose [46]	CNOS [43]	0.338	0.469	0.452
Co-op [41]	CNOS [43]	**0.592**	**0.642**	**0.626**
Ours	CNOS [43]	0.190	0.476	0.504

Table 5. **Model-Based 6D Localization of Unseen Objects without Refinement.** We report Average Recall (AR) on 3 BOP datasets and compare with current SOTA model-based RGB based pose estimation methods *without refinement*. We use the default CNOS [43] detections provided by the BOP challenge when indicated. Data taken from [41].

Model-Based Detection. AP for amodal detection on T-LESS, TUD-L and YCB-V datasets is reported in Tab. 3. We achieve state-of-the-art results on TUD-L, outperform-

Figure 5. **Object Surface Reconstruction and Camera Pose Estimation on Unseen Objects.** We show object surface points and camera poses as predicted by the decoder based on four images of randomly chosen objects in different scenarios: (Left) A static coffee machine standing on a table, captured by a dynamic camera. (Middle) A label machine in different environments, including occlusions. (Right) An espresso mug manipulated in hand, captured by a static camera. In all three scenarios, our model is able to predict object-centric camera poses and surface points. We map RGB pixel color to corresponding 3D point to show correct 2D-3D mapping. Blue is the anchor image.

ing the previous best method by 3pp. On YCB-V we would rank 6th out of 13 methods reported on the BOP leaderboard. On T-LESS we achieve a precision of 0.183, which is probably due to the strong similarities between the objects, the lack of texture as well as the dataset containing many cluttered scenes with objects of the same instance. All these factors are not present in our training data.

Model-Based Segmentation. In addition to detection we also report object segmentation AP on the three datasets. We report the results in Tab. 4. Compared to other networks our method is less precise on pixel level, which could be due to border artifacts as a results of patch-scaling. As with the detection results, the precision on T-LESS is low, which we attribute to the same reasons as mentioned above.

Model-Based Refiner-Free 6DoF Pose Estimation. To evaluate the 6DoF Pose Estimation capabilities of our method independent of the detection quality we report the *average recall AR* on T-LESS, TUD-L and YCB-V with default CNOS [43] detection and *without additional refinement step* in Tab. 5 as is standard practice in the literature. Although our network was not designed for the model-based category we achieve high results in TUD-L and YCB-V, while only Co-op [41] achieves better results. Our method fails to handle the symmetric and textureless objects in T-LESS, which is reflected on the low average recall of 0.190. In addition we report the results of our method using ground truth and NeMO detections. Surprisingly, although the NeMO detections show higher precision than the CNOS detections as reported in Tab. 3, the average recall on pose estimation task is lower when using NeMO detections. This is due to the difference in how average precision and average recall are evaluated. For results on model-based pose estimation with refinement we refer to Supplementary Tab. 9.

# NeMO Points	AP ↑	AP_{MSPD} ↑	AP_{MSSD} ↑	Time/Image (s) ↓
10	0.004	0.005	0.002	0.402
50	0.112	0.130	0.094	0.671
100	0.214	0.231	0.196	0.460
200	0.290	0.297	0.282	0.393
500	0.380	0.379	0.381	0.340
1000	0.383	0.389	0.376	0.320
1500	0.378	0.37	0.385	0.336

Table 6. **Number of NeMO Input Points vs AP.** We report the AP on YCB-V 6D pose estimation with ground truth detections by varying number of randomly sampled input points.

4.3. Analyzing the NeMO Representation

In this section we analyze the properties of the NeMO representation. For all experiments we report AP of 6DoF pose estimation on the test split of YCB-V with real template images randomly chosen from the public real training set provided by the BOP benchmark [25]. We use ground truth detections and no refinement unless stated otherwise. A smaller decoder that only outputs pointmap and confidence is used in this section. Additional experiments and analyses can be found in Supplementary Sec. 7.6.

Varying number of template images. We show the relation between the number of template images used to generate a NeMO and its influence on 6DoF pose estimation and the required memory footprint in Fig. 6. It shows that while more template images enhance precision, acceptable performance is already achieved with just three views. We emphasize the fact that whereas the number of templates increases, the runtime and memory consumption of our decoder model stays quasi constant while the precision increases. Since we generate our NeMOs before the inference task, we shift the computational heavy part of attending all template features with each other to the offline phase.

Varying number of NeMO Points. Although the model was trained on a fixed sized number of input points $\mathcal{Q}$ we show in Tab. 6 that the encoder and decoder can adapt to

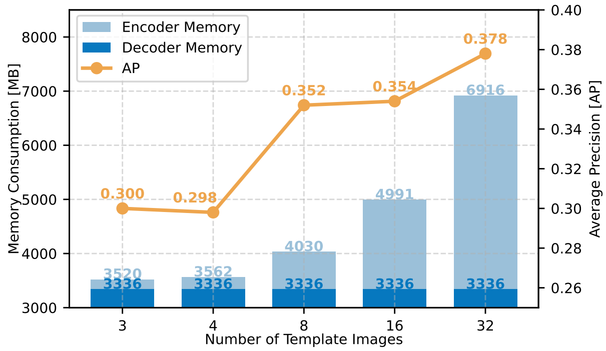

Figure 6. **Memory consumption and AP on 6DoF Pose Estimation vs. Number of template images.** While the offline NeMO generation requires more memory as the number of template images increases, the inference memory remains constant. Additionally, more template images increase the AP on 6DoF Pose Estimation on YCB-V with ground truth detections.

different point cloud sizes, allowing for a dynamic adaptation of the models memory consumption and precision. We observe that the precision increases as the number of NeMO points increases up to 500 where it stagnates around 0.38. A visualization of NeMO features can be seen in Fig. 2.

Transforming NeMO coordinate system. During training, we randomly transform our NeMO point cloud before passing it to the decoder. In this section we evaluate if the decoder adapts its output based on the positions of the NeMO points. To test the rotation-equivariance between the NeMO point cloud and the predicted pointmaps X of the template images, we rotate the NeMO point cloud around the z-axis in 10 degree steps and evaluate the Chamfer distance [15] between the predicted pointmap and the ground truth CAD model rotated by the same angle. As can be seen in Supplementary Fig. 11, the Chamfer distance [15] between the pointmaps and the ground truth CAD model remains low while the Chamfer distance between a non-rotating pointmap varies, indicating that the pointmap prediction is rotating accordingly. This is an interesting property of our network for future work, in which parts of the NeMO could be transformed online during inference.

Extending NeMO. As the NeMO representation is based on a set of points, it can easily be extended. To see if the addition of new points from a different NeMO of the same object leads to better results, we combine two NeMOs with the same anchor image. In Supplementary Tab. 10 we show the Chamfer distance between the predicted pointmap and the ground truth CAD model for the original NeMO and the extended one, observing that the Chamfer distance is lower for the extended NeMO than for the original one. This shows that we can combine two sets of NeMO points without disturbing the decoder while improving its performance. This is beneficial in scenarios, where the hardware is memory limited and thus, fewer template images can be used.

5. Limitations

Despite promising results, our method exhibits several limitations. The encoder is trained to predict surface points, which leads to difficulties with symmetric objects, as demonstrated on the T-LESS dataset. We attribute this not to the pointmap representation itself, but to limitations in the current training procedure; addressing this will require further research. Additionally, the encoder performs poorly on highly textureless objects, likely due to their underrepresentation in the training set. Future work will focus on scaling the dataset to include a broader and more diverse set of objects. Another limitation is that the encoder does not directly predict bounding boxes; instead, segmentation masks are used as a proxy. This can result in merged bounding boxes when multiple instances of the same object are present—an issue we plan to resolve in future work.

6. Conclusion

In this work, we presented an encoder and decoder architecture for Neural Memory Object (NeMO), a general and versatile object-centric representation that can be used for few-shot perception tasks such as object detection, segmentation and pose estimation using only an unordered set of RGB images. Through thorough experiments, we demonstrated that our method can be used for few-shot, model-free and model-based unseen perception task, partially outperforming state-of-the-art methods. Our approach differs to alternative methods by (i) being capable to incorporate information from multiple RGB recorded images into a single representation without any camera parameters, (ii) utilizing a single network for multiple perception tasks, (iii) having constant inference time and memory requirements regardless of the number of template images, (iv) predicting amodal segmentation masks without CAD-model, (v) being able to incorporate CAD-model information if given, (vi) allowing dynamically changing memory usage and precision, based on hardware capabilities, (vii) outsourcing the object's information from the network weights, allowing for quick adaptation to novel objects without retraining. Furthermore, we contribute a realistic large-scale and balanced object-centric dataset that we deem beneficial for the broader research community. Future work includes combining multiple NeMOs for articulated objects. Additionally, we strive to increase robustness of the applied task-specific encoder w.r.t. symmetrical and textureless objects as motivated by the results on the T-LESS dataset. We hope this work stimulates discussion on decoupling object knowledge from model weights and helps advance model-free few-shot perception for unseen objects.

References

[1] Philipp Ausserlechner, David Haberger, Stefan Thalhammer, Jean-Baptiste Weibel, and Markus Vincze. ZS6D: Zero-shot 6D Object Pose Estimation using Vision Transformers. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 463–469, 2024. 3, 6

[2] BOP Authors. Model-based unseen object 2d detection leaderboard bop-classic. https://bop.felk.cvut.cz/leaderboards/detection-unseen-bop23/bop-classic-core/, . Accessed: 2025-07-30. 6

[3] BOP Authors. Model-based unseen object 6d detection leaderboard bop-classic. https://bop.felk.cvut.cz/leaderboards/pose-detection-unseen-bop24/bop-classic-core/, . Accessed: 2025-07-30. 14

[4] BOP Authors. Model-based unseen object 2d segmentation leaderboard bop-classic. https://bop.felk.cvut.cz/leaderboards/detection-unseen-bop23/bop-classic-core/, . Accessed: 2025-07-30. 6

[5] BOP Authors. Model-free unseen object 2d detection leaderboard bop-h3. https://bop.felk.cvut.cz/leaderboards/modelfree-detection-unseen-bop24/bop-h3/, . Accessed: 2025-07-30. 5

[6] BOP Authors. Model-free unseen object 6d detection leaderboard bop-h3. https://bop.felk.cvut.cz/leaderboards/modelfree-pose-detection-unseen-bop24/bop-h3/, . Accessed: 2025-07-30. 5

[7] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 6

[8] Andrea Caraffa, Davide Boscaini, Amir Hamza, and Fabio Poiesi. Freeze: Training-free zero-shot 6d pose estimation with geometric and vision foundation models. In *European Conference on Computer Vision (ECCV)*, 2024. 14

[9] Jianqiu Chen, Zikun Zhou, Mingshan Sun, Rui Zhao, Liwei Wu, Tianpeng Bao, and Zhenyu He. Zeropose: Cad-prompted zero-shot object 6d pose estimation in cluttered scenes. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(2):1251–1264, 2025. 3

[10] 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. *arXiv preprint arXiv:2212.08051*, 2022. 5

[11] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023. 1, 5

[12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 337–33712, Salt Lake City, UT, USA, 2018. IEEE. 2

[13] A. Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[14] 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 *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560, 2022. 5

[15] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2463–2471, 2017. 8

[16] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 4

[17] Max Gandyra, Alessandro Santonicola, and Michael Beetz. Noctis: Novel object cyclic threshold based instance segmentation, 2025. 2, 6

[18] Anas Gouda and Moritz Roidl. Dounseen: Tuning-free class-adaptive object detection of unseen objects for robotic grasping, 2023. 5

[19] Andrew Guo, Bowen Wen, Jianhe Yuan, Jonathan Tremblay, Stephen Tyree, Jeffrey Smith, and Stan Birchfield. HANDAL: A dataset of real-world manipulable object categories with pose annotations, affordances, and reconstructions. In *IROS*, 2023. 5

[20] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004. 4

[21] Xingyi He, Jiaming Sun, Yuang Wang, Di Huang, Hujun Bao, and Xiaowei Zhou. Onepose++: Keypoint-free one-shot object pose estimation without CAD models. In *Advances in Neural Information Processing Systems*, 2022. 2

[22] Yisheng He, Yao Wang, Haoqiang Fan, Jian Sun, and Qifeng Chen. FS6D: Few-Shot 6D Pose Estimation of Novel Objects. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6804–6814, New Orleans, LA, USA, 2022. IEEE. 2

[23] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, Caner Sahin, Fabian Manhardt, Federico Tombari, Tae-Kyun Kim, Jiri Matas, and Carsten Rother. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 5

[24] Tomáš Hodaň, Pavel Haluza, Štěpán Obdržálek, Jiří Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017. 5

[25] Tomáš Hodaň, Martin Sundermeyer, Yann Labbé, Van Nguyen Nguyen, Gu Wang, Eric Brachmann, Bertram Drost, Vincent Lepetit, Carsten Rother, and Jiří Matas. BOP Challenge 2023 on Detection, Segmentation and Pose Estimation of Seen and Unseen Rigid Objects. *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024. 1, 5, 7

[26] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7, 2020. 5

[27] Hanwen Jiang, Zhenyu Jiang, Yue Zhao, and Qixing Huang. Leap: Liberate sparse-view 3d modeling from camera poses. *ArXiv*, 2310.01410, 2023. 3, 4

[28] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 2, 6

[29] 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 *Proceedings of The 6th Conference on Robot Learning*, pages 715–725. PMLR, 2023. 2, 3, 6

[30] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem, 2009. 4

[31] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding Image Matching in 3D with MASt3R. In *Computer Vision – ECCV 2024*, pages 71–91. Springer Nature Switzerland, Cham, 2025. 2, 4

[32] Amy Lin, Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. RelPose++: Recovering 6D Poses from Sparse-view Observations. In *2024 International Conference on 3D Vision (3DV)*, pages 106–115, 2024. ISSN: 2475-7888. 2

[33] Jiehong Lin, Lihua Liu, Dekun Lu, and Kui Jia. Sam-6d: Segment anything model meets zero-shot 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27906–27916, 2024. 3, 6

[34] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024. 2

[35] Xingyu Liu, Gu Wang, Chengxi Li, Yingyue Li, Chenyangguang Zhang, Ziqin Huang, and Xiangyang Ji. Gfreedet: Exploiting gaussian splatting and foundation models for model-free unseen object detection in the bop challenge 2024. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), CV4MR Workshop*, 2025. 2, 5

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 5

[37] Yangxiao Lu, Jishnu Jaykumar P, Yunhui Guo, Nicholas Ruozzi, and Yu Xiang. Adapting pre-trained vision models for novel instance detection and segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024. 2, 6

[38] 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, 2020. 2

[39] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, 2016. 5

[40] Sungphill Moon, Hyeontae Son, Dongcheol Hur, and Sangwook Kim. Genflow: Generalizable recurrent flow for 6d pose refinement of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10039–10049, 2024. 6

[41] Sungphill Moon, Hyeontae Son, Dongcheol Hur, and Sangwook Kim. Co-op: Correspondence-based novel object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11622–11632, 2025. 6, 7, 14

[42] Van Nguyen Nguyen, Yuming Du, Yang Xiao, Michaël Ramamonjisoa, and Vincent Lepetit. PIZZA: A Powerful Image-only Zero-Shot Zero-CAD Approach to 6 DoF Tracking. In *2022 International Conference on 3D Vision (3DV)*, pages 515–525, 2022. 2

[43] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. CNOS: A Strong Baseline for CAD-based Novel Object Segmentation. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2126–2132, Paris, France, 2023. IEEE. 2, 5, 6, 7

[44] Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. GigaPose: Fast and Robust Novel Object Pose Estimation via One Correspondence. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9903–9913, Seattle, WA, USA, 2024. IEEE. 2, 3, 6

[45] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 5

[46] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan. Foundpose: Unseen object pose estimation with foundation features. In *European Conference on Computer Vision*, pages 163–182. Springer, 2024. 6

[47] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. LatentFusion: End-to-End Differentiable Reconstruction and Rendering for Unseen Object Pose Estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10707–10716, Seattle, WA, USA, 2020. IEEE. 2

[48] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12188, 2021. 4, 5

[49] Iqbal H. Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3):160, 2021. Epub 2021 Mar 22. PMID: 33778771. PMC7983091. Accessed 2025-07-29. 3

[50] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching With Graph Neural Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4937–4946, Seattle, WA, USA, 2020. IEEE. 2

[51] Jingnan Shi, Rajat Talak, Harry Zhang, David Jin, and Luca Carlone. Crisp: Object pose and shape estimation with test-time adaptation. *arXiv preprint arXiv:2412.01052*, 2024. 2

[52] Ivan Shugurov, Fu Li, Benjamin Busam, and Slobodan Ilic. OSOP: A Multi-Stage One Shot Object Pose Estimation Framework. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6825–6834, New Orleans, LA, USA, 2022. IEEE. 3

[53] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-Free Local Feature Matching with Transformers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8918–8927, Nashville, TN, USA, 2021. IEEE. 2

[54] 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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6815–6824, New Orleans, LA, USA, 2022. IEEE. 2

[55] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[56] Stephen Tyree, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Jeffrey Smith, and Stan Birchfield. 6-dof pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark. In *International Conference on Intelligent Robots and Systems (IROS)*, 2022. 5

[57] Maximilian Ulmer, Wout Boerdijk, Rudolph Triebel, and Maximilian Durner. Conditional latent diffusion models for zero-shot instance segmentation, 2025. 2

[58] Unknown. Method: Frtpose.v1 (sam6d-fastsam). https://bop.felk.cvut.cz/method_info/784/, 2024. Accessed: 2025-07-30. 1

[59] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 4

[60] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20697–20709, 2024. 4, 12

[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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 606–617, Vancouver, BC, Canada, 2023. IEEE. 2

[62] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17868–17879, Seattle, WA, USA, 2024. IEEE. 2

[63] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Liang Pan Jiawei Ren, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5

[64] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018. 5

[65] Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. RelPose: Predicting Probabilistic Relative Rotation for Single Objects in the Wild. In *Computer Vision – ECCV 2022*, pages 592–611. Springer Nature Switzerland, Cham, 2022. 2

[66] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything, 2023. 2, 6