INSTANTREFINE: ONE-REFERENCE 6D POSE REFINEMENT FOR NOVEL OBJECTS BY DIRECT REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

The task of 6D object pose estimation plays an important role in downstream robotics and augmented reality applications. To broaden its practical applicability, recent research increasingly leverages a single reference view rather than requiring a complete 3D object model. A common paradigm first estimates a coarse pose and subsequently refines it using a dedicated module. However, existing refiners primarily focus on establishing fine-grained 3D–3D correspondences, where pose updates are derived indirectly and remain susceptible to suboptimal predictions due to the misalignment between correspondence metrics and pose accuracy. To address this, we propose InstantRefine, a simple yet effective plug-and-play refiner that directly regresses pose updates. By representing both the coarse pose and the observations as a set of point tokens within a Transformer architecture, our method bypasses explicit correspondence matching, 3D model reconstruction, and geometric pose solvers (e.g., RANSAC/SVD) entirely. Beyond the one-reference setting, our framework naturally extends to multi-view references, significantly enhancing pose accuracy. Extensive experiments across multiple benchmarks demonstrate that InstantRefine consistently outperforms existing refiners under diverse pose initializations and extreme viewpoint variations.

1 INTRODUCTION

The 6-DoF (Degrees of Freedom) object pose estimation problem aims to recover the rotation and translation of the target object from visual observations. It is a fundamental problem in computer vision and has wide applications in augmented reality as well as robotic manipulation. Traditional model-based methods (Xiang et al., 2018; Peng et al., 2019; Liu et al., 2025c) focus on pose estimation for a predefined set of objects. While accurate, they require extensive training on instance-specific annotations and inherently fail to generalize to novel, unseen objects. To bypass instance-specific training, subsequent zero-shot methods (Labbé et al., 2022; Wen et al., 2024; Nguyen et al., 2024) leverage complete 3D models (e.g., CAD models) as reference templates. However, obtaining precise 3D CAD models in unconstrained real-world environments is often impractical or cost-prohibitive, severely restricting their practical deployment.

To eliminate the dependency on 3D models, recent research has shifted towards model-free paradigms (Sun et al., 2022; He et al., 2022a), particularly one-reference pose estimation (Corsetti et al., 2024; Liu et al., 2025a;b; Zuo et al., 2026). Given only one reference view of a novel object, these methods estimate its relative pose in a query view without any prior 3D model or test-time re-training. In general, this problem is formulated as relative pose estimation. Existing methods assume that depth is given and compute the pose by estimating 3D–3D correspondences. For example, coordinate-map-based methods (Liu et al., 2025a; Zuo et al., 2026) encode 3D–3D correspondences via coordinate maps, and point-cloud-registration-based methods (Liu et al., 2025b; Che et al., 2026) explicitly establish matching point pairs using learned descriptors.

To turn a coarse pose estimate into a high-precision prediction, pose refinement modules are widely adopted as an essential post-processing or built-in stage. However, conventional refinement architectures largely mimic the coarse-stage designs by densifying point clouds to predict finer correspondences. We identify two major limitations in this correspondence-based refinement paradigm. First, *objective misalignment*—the network is trained to minimize correspondence errors rather than pose errors. This misalignment is problematic in practice: two correspondence sets may have the same

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

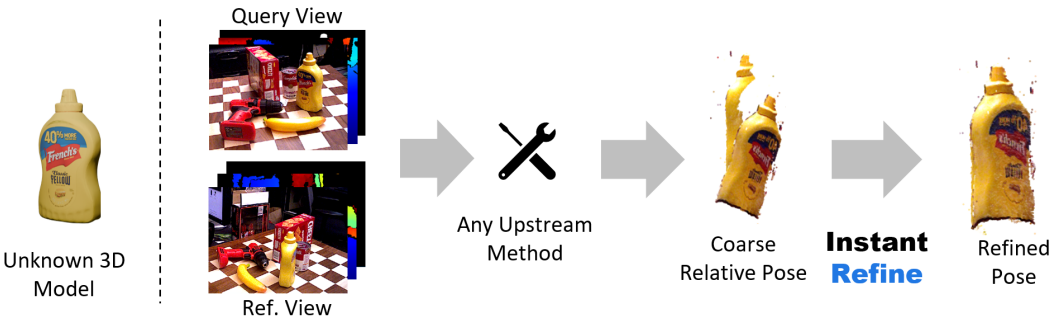

Figure 1: Illustration of one-reference object pose refinement. Given a single reference view and a query view with an initial coarse relative pose, the goal is to refine the query pose without knowing the complete 3D model.

correspondence error yet lead to completely different poses (Wang et al., 2021). Second, *limited spatial overlap*—under large viewpoint variations, as the viewpoint difference between two views grows, the overlap region shrinks significantly, leading to spurious descriptor matches that severely degrade downstream pose optimization.

To address these limitations, we propose InstantRefine, a novel plug-and-play refiner that performs direct pose regression for one-reference 6D pose estimation. Instead of establishing explicit correspondences or solving PnP/SVD problems, InstantRefine transforms visual observations from both views into a Normalized Anchor Frame (NAF) established by the initial coarse pose and represents them as a set of point tokens. A Transformer (Vaswani et al., 2017) architecture then directly regresses the 6D delta pose from these unified spatial tokens. By learning pose updates end-to-end, InstantRefine completely eliminates explicit descriptor matching, 3D model reconstruction, and geometric solvers. Furthermore, our framework seamlessly extends to multi-view reference settings, unlocking higher refinement accuracy without requiring explicit 3D mesh generation.

Our main contributions are summarized as follows: 1) We propose InstantRefine, a simple yet effective one-reference 6D pose refiner that directly regresses pose updates, bypassing explicit 3D model reconstruction, descriptor matching, and geometric pose solvers; 2) We introduce a unified Transformer-based architecture over point tokens anchored in a reference frame, which naturally scales to multi-view reference settings; 3) Extensive evaluations across LINEMOD, YCB-V, and TUD-L demonstrate that InstantRefine consistently achieves state-of-the-art performance across diverse pose initializations and extreme viewpoint variations.

2 Related Works

6D Object Pose Estimation. Traditionally, 6D object pose estimation methods operate under the assumption that a precise 3D model, such as a CAD model, is available for each target object. Although model-based methods (Xiang et al., 2018; Su et al., 2022; Wang et al., 2021; Liu et al., 2025c) demonstrate high accuracy (Sundermeyer et al., 2023), they are challenging to apply to novel objects due to training with absolute poses and pre-defined 3D models. Subsequent methods (Labbé et al., 2022; Nguyen et al., 2024) handle novel objects by estimating the relative pose between observations and templates, still requiring a 3D model for template rendering. In contrast, model-free methods aim to estimate the pose of an object without relying on its pre-given 3D model during inference. Some approaches (Sun et al., 2022; He et al., 2022a; Wen et al., 2024; Liu et al., 2022) first reconstruct the object before pose estimation. A major limitation of these methods is their dependence on both sufficient viewpoints and accurate pose annotations, which are often unavailable or expensive to obtain. Other methods bypass reconstruction and instead compare the query view with multiple known views (He et al., 2022b) or a single reference view (Corsetti et al., 2024; Liu et al., 2025b;a; Zuo et al., 2026) of the object. Among these methods, one-reference methods have gained increasing attention due to their practicality. Requiring only one reference image, they essentially perform relative pose estimation and thus do not require the reference view's absolute pose to be known. While effective, these approaches rely on estimated correspondences as a surrogate

for pose optimization rather than directly optimizing the pose. However, the surrogate objective for correspondence regression does not necessarily align with the actual pose error after optimization, as argued in GDR-Net (Wang et al., 2021). As a result, the estimated pose may be suboptimal, and further refinement can be effective for higher precision.

6D Object Pose Refinement. In literature, a post refinement is necessary to acquire high-precision 6D pose. When 3D models are available for rendering, 6D objects pose refinement is traditionally addressed by a refinement module with a render&compare loop, which aims to refine the estimated pose by rendering a synthetic image iteratively until the rendered image aligns pixel-wisely with the observed real-world image. For example, GenFlow (Moon et al., 2024) predicts dense optical flow between the two images and uses a cascaded architecture containing convolutional GRUs (Cho et al., 2014) and a differentiable PnP (Fischler & Bolles, 1981) solver to update the 6D pose step-by-step in a coarse-to-fine manner. MegaPose (Labbé et al., 2022) introduces a 6D pose refiner based on render&compare strategy, which regresses the delta pose by comparing the two images. FoundationPose (Wen et al., 2024) trains a similar refiner which achieves strong generalizability via large-scale synthetic training. However, these refiners require reconstructing a complete 3D model for rendering, which is not trivial to obtain from a single reference view. Current reconstruction-free pose refinement approaches for single or multiview reference typically treated the problem as a conditional point cloud registration problem, which aims to align two point clouds by finding the optimal rigid transformation between them given the initial poses. For instance, FS6D (He et al., 2022b) use ICP (Besl & McKay, 1992) for ierative refinement. However, ICP is known to be sensitive to initialization and prone to local minima. Another solution is to obtain fine-grained dense correspondences, as done by the refinement modules in UNOPose (Liu et al., 2025b) and COG (Che et al., 2026), which establish fine-grained correspondences based on the coarse pose from a previous stage.

3 METHODOLOGY

3.1 PROBLEM STATEMENT

Given only a single reference view of a target object, one-reference pose estimation aims to infer the relative object pose in a query view. Our goal is to address the problem of refining a coarse 6D pose provided by an external method into an accurate pose. Formally, the inputs to our method include:

- A query RGB-D image $\mathcal{I}_q = (\mathcal{C}_q, \mathcal{D}_q) \in \mathbb{R}^{H\times W\times 4}$ with known intrinsics captured from an unknown viewpoint, consisting of color $\mathcal{C}_q \in \mathbb{R}^{H\times W\times 3}$ and depth $\mathcal{D}_q \in \mathbb{R}^{H\times W\times 1}$.
- A reference RGB-D image $\mathcal{I}_r = (\mathcal{C}_r, \mathcal{D}_r) \in \mathbb{R}^{H\times W\times 4}$ with known intrinsics.
- Binary object masks $\mathcal{M}_r \in \mathbb{R}^{H\times W}$ and $\mathcal{M}_q \in \mathbb{R}^{H\times W}$ for the reference and query images, respectively.
- An initial pose $\mathbf{T}^0_{rq}$ given by an external upstream pose estimation method.

Our method outputs a pose update that produces a new pose estimate, and the updated pose can then be fed back into the same refiner for the next iteration. Specifically, at the t-th refinement iteration, the refiner predicts a pose update $\Delta\mathbf{T}^t$ from the current pose estimate $\mathbf{T}^{t-1}_{rq}$ and the observations. The pose is then updated as

$$\mathbf{T}^t_{rq} = \Delta\mathbf{T}^t\mathbf{T}^{t-1}_{rq}. \tag{1}$$

Here, $\mathbf{T}^0_{rq}$ denotes the initial pose provided by the upstream method, and $\mathbf{T}^t_{rq}$ for $t > 0$ denotes the pose after the t-th refinement iteration. This process can be repeated until a stopping criterion is met, such as reaching a maximum number of iterations or observing a sufficiently small pose update. Note that our method refines the relative object pose between two views, which means that the absolute pose of the reference is not necessary for the model. It is only needed when we want to calculate the absolute pose of the query with respect to a predefined object frame.

3.2 OVERVIEW

The overview of our method is presented in Fig. 2. First, the point cloud in each view is back-projected and sampled from the depth images and transformed in to a normalize anchor frame. Then,

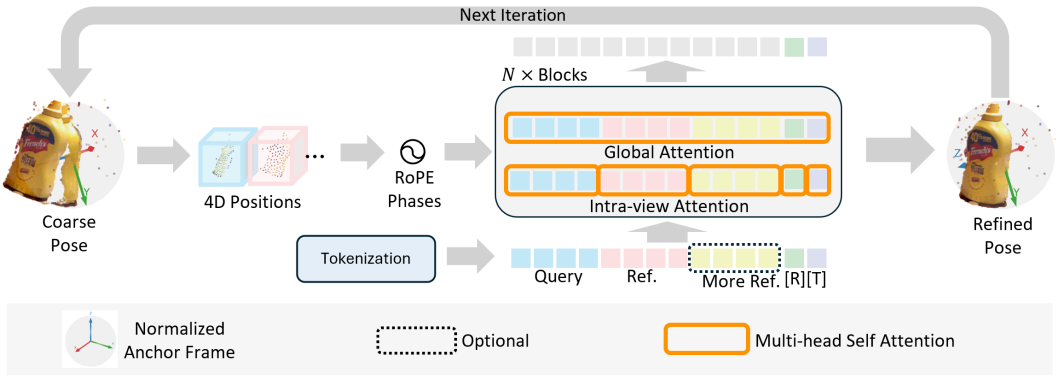

Figure 2: Overview of InstantRefine. To encode relative positional information of the coarse pose, we transform the observations into a normalized anchor frame (NAF). Each poin token is assigned with a 4D position associated with NAF when processed by RoPE. The global information for rotation and translation is collected by the [R] and [T] tokens in the Transformer blocks respectively.

each view is tokenized independently and the resulting point tokens are processed by a Transformer. To encode global positional information, each token is assigned with a 4D position in the normalize anchor frame. Finally, the delta rotation and translation are regressed from the Transformer. The refined pose can be further processed by the same network in the next iteration.

3.3 Pose Refinement Network

Normalized Anchor Frame (NAF). Our refinement is performed in a normalized local coordinate frame centered on the reference object, which we refer to as the normalized anchor frame. The origin of the NAF is defined at the center of the reference point cloud. To account for size variations, the points are rescaled by a diameter factor derived from both the area of the reference mask and the camera intrinsics. Then we transform the query points into the NAF using the initial relative pose. Formally, the reference point cloud is derived by:

$$\mathcal{P}_r = \mathbf{N}\left(\mathcal{S}\left(\Pi^{-1}(\mathcal{D}_r)\left[\mathcal{M}_r = 1\right], M\right)\right), \tag{2}$$

where $\Pi^{-1}(\cdot)$ is the back-projection operator, $\mathcal{S}(\cdot)$ is the sampling operator, M is the number of points, and $\mathbf{N}$ represents the normalization transformation, which can be derived from the input (details in the supplementary). The query points in the NAF is calculated by:

$$\mathcal{P}_q^0 = \mathbf{N}\mathbf{T}_{rq}^0\left(\mathcal{S}\left(\Pi^{-1}(\mathcal{D}_q)\left[\mathcal{M}_q = 1\right], M\right)\right), \tag{3}$$

After this transformation, the points from both views and their initialized relative pose are all represented in a unified coordinate frame, facilitating further spatial perception for neural networks. In NAF, since $\mathbf{T}_{rq}^0$ is typically not accurate, $\mathcal{P}_r$ and $\mathcal{P}_q^0$ are not well-aligned to form a whole object. To align them, we need to apply a delta pose on $\mathcal{P}_q^0$. The target delta pose of the neural network $[\Delta\bar{\mathbf{T}}]_{\mathbf{N}}$ can be derived as

$$[\Delta\bar{\mathbf{T}}]_{\mathbf{N}} = \mathbf{N}\bar{\mathbf{T}}_{rq}(\mathbf{T}_{rq}^0)^{-1}\mathbf{N}^{-1}, \tag{4}$$

where $\bar{\mathbf{T}}_{rq}$ is the ground-truth relative pose in the world frame, and $[\cdot]_{\mathbf{N}}$ denotes the transformation expressed in NAF. To parameterize $[\Delta\bar{\mathbf{T}}]_{\mathbf{N}}$, we decouple the translation target from rotation by directly regressing the translation component of its matrix inverse. Detailed formulations and derivations are provided in Appendix Sec. A.1.

Tokenization. As illustrated in Fig. 3, to enable Transformer-based processing, each view is first tokenized by sampling a sparse set of keypoints. To construct informative point tokens, we combine features across multiple representations. Specifically, local geometric features are captured using a mini PointNet (Qi et al., 2017a;b) within an SE(3)-invariant local reference frame (LRF) (Liu et al., 2025b). To enrich spatial semantics and appearance details, these geometric features are further augmented with deep representations from DINOv2 (Oquab et al., 2023) and SD-VAE (Rombach et al., 2022), providing rich semantic grounding and textural cues for texture-rich objects, respectively.

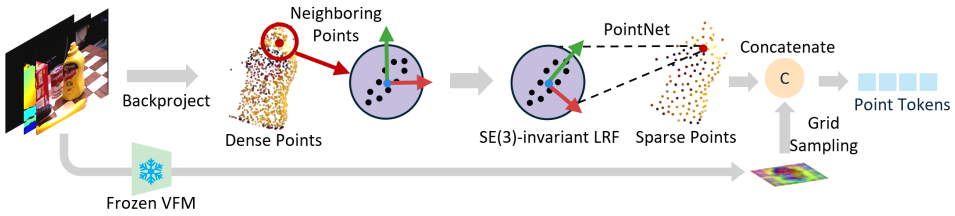

Figure 3: Tokenization in our method. Each view is converted to a fixed-size token set independently. We use farthest point sampling (FPS) to sample point clouds at different densities. Each token aggregates the information from its neighboring points in dense points. The neighboring points are transformed into a local reference frame (LRF) to ensure SE(3) invariance before processed by the mini PointNet. The features from different sources are concatenated along the feature dimension and then projected to a fixed number of dimensions.

4D Relative Positional Encoding (RPE). While tokenization captures local geometry and semantics, it lacks global positional awareness. We therefore introduce relative positional encoding to incorporate global positional information into the tokens. Specifically, each token is assigned a 4D positional coordinate (role, x, y, z), where role is an enumeration with only three values, 0, 1, and 2, corresponding to the rotation token, translation token, and point token, respectively. For the rotation and translation tokens, their x, y, and z coordinates are padded with their respective role values, similar to the position IDs for textual inputs as in Qwen2.5-VL (Bai et al., 2025). For point tokens, the x, y, and z coordinates are set to the coordinates of the corresponding point in NAF. These coordinates are then utilized by RoPE (Su et al., 2024). Here, for each token, its coordinate is encoded into a rotation matrix, which is applied to the query and key vectors in the attention mechanism. In this way, the attention score between two tokens depends not only on their content but also on their relative position and role difference. For point tokens, the relative spatial relationship is captured by the differences in (x, y, z), while the role dimension allows the model to distinguish rotation and translation tokens from point tokens. This design enables the Transformer to reason about both geometry and token functionality during information exchange. To demonstrate its effectiveness, we compare the 4D relative positional encoding with absolute positional encoding in Sec. 4.3.

Pose Regression. We concatenate the point tokens from different views into a single token set and append two learnable task-specific tokens—a rotation token $[R]$ and a translation token $[T]$—to aggregate global context for decoupled pose targets. A Transformer architecture with alternating intra-view and global attention mechanisms is then employed to perform layer-wise information exchange among all tokens. Through this feature aggregation, $[R]$ and $[T]$ independently gather spatial and contextual cues tailored for rotation and translation, respectively. Finally, two independent linear heads predict the delta rotation and delta translation from these two decoupled tokens separately. As an alternative baseline, one could derive global pose estimates by pooling point tokens within each view prior to regression; we evaluate this design choice in our ablation study.

Extend to Multiview Reference. Our method can naturally extend to multi-view reference settings. The point tokens from additional reference views are appended to the token set before feeding them into the transformer. For position encoding, we designate an arbitrary reference view to establish the NAF and then transform the points from all other reference views into this coordinate frame. During the intra-view attention, tokens within each extra reference view perform self-attention independently.

4 Experiments

4.1 Experimental Setup

Training Data. Following previous works (Liu et al., 2025a; Zuo et al., 2026), we utilize large-scale synthetic datasets for training, including the FoundationPose dataset (Wen et al., 2024) and the OV9D dataset (Cai et al., 2024). The FoundationPose dataset (Wen et al., 2024) contains scenes rendered using objects from GSO (Downs et al., 2022) and Objaverse (Deitke et al., 2023). To ensure sufficient viewpoints of each object, we include the single-object rendering subset of OV9D (Cai

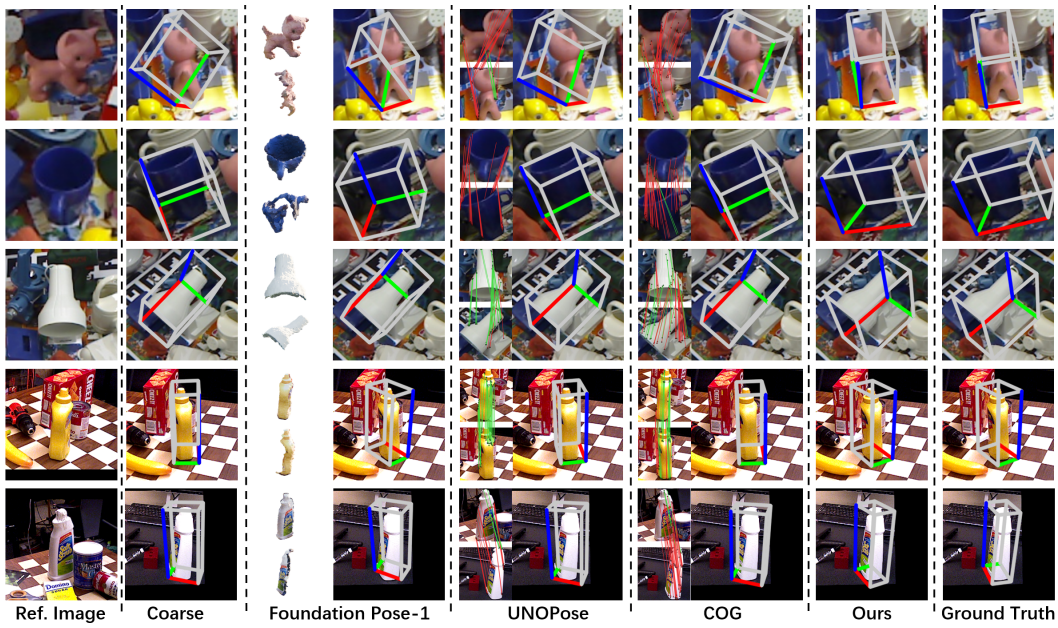

Figure 4: Qualitative comparison. Poses are visualized as bounding boxes with distinctively colored edge. For FoundationPose, we presents the reconstructed 3D model from FoundationPose using a single reference view. For UNOPose and COG, the correct and wrong correspondences are colored with green and red respectively.

et al., 2024). We randomly sample viewpoint pairs and add noise to the relative pose to simulate the coarse pose provided by the upstream method. The noise is added by following CIR (Lipson et al., 2022).

Benchmark Datasets. Our method is evaluated on several benchmark datasets commonly used for one-reference pose estimation, including LINEMOD (Hinterstoisser et al., 2011), YCB-V (Xiang et al., 2018) and TUD-L (Hodaň et al., 2018). LINEMOD contains textureless objects with large viewpoint variations. YCB-V features heavy occlusions and cluttered scenes with texture-rich objects. TUD-L consists of three objects captured under diverse lighting conditions. Following prior works (Liu et al., 2025a; Zuo et al., 2026), for LINEMOD and YCB-V, the first view of each object in the entire dataset is used as the reference view; On TUD-L dataset, we use the same view selected by UNOPose as the reference.

Evaluation Metrics. We report the ADD(-S) and the AR (Hodaň et al., 2018) metrics. These metrics first compute the recall rate with different types of errors between predictions and ground truth. For a detailed description of the error calculation, we refer the reader to (Hodaň et al., 2018; 2016).

4.2 Comparison with State-of-the-art Methods

Our pose initializations are taken from the state-of-the-art one-reference methods, including the coordinate-map-based method CoordAR (Zuo et al., 2026) and the point-based methods UNOPose (Liu et al., 2025b) and COG (Che et al., 2026).

Compared Methods. For single reference refinement, we implement ICP (Besl & McKay, 1992) based refinement as a baseline, in which the pose is updated iteratively by establishing correspondences between the reference depth and the query depth and minimizing the point-to-point distance. To demonstrate superiority over correspondence-based methods, our method is compared with the refinement stage proposed by UNOPose and COG, both of which refine the correspondences based on the coarse pose. To compare with reconstruction-based methods, we reconstruct a 3D mesh for each object using a one-reference image by FoundationPose following One2Any (Liu et al., 2025a) and the coarse pose is subsequently refined by the render-and-compare loop of FoundationPose.

Table 1: Comparison of different pose refiner methods on LINEMOD, YCB-Video, and TUD-L datasets. Our method is built upon various coarse pose inputs. We report the ADD(-S) (%) and the AR (%). The best results per coarse source are in **bold**. **RF** denotes reconstruction-free.

Initialization	Refiner	RF	Ref.	Iter.	LINEMOD		YCB-Video		TUD-L	
					ADD(-S)	AR	ADD(-S)	AR	ADD(-S)	AR
CoordAR	No refine	–	–	–	78.2	68.3	74.1	59.5	66.0	56.3
	FoundationPose	✗	16	5	92.3	86.5	83.4	77.0	76.5	69.5
	ICP-Recon	✗	16	$\leq$30	81.7	72.3	64.9	57.3	66.8	64.5
	Ours	✓	16	5	**93.5**	**87.2**	**88.8**	**81.4**	**80.3**	**73.3**
	FoundationPose-1st	✗	1	5	74.5	69.2	70.8	62.7	40.1	43.8
	ICP-Depth	✓	1	$\leq$30	71.0	63.9	60.8	52.3	54.0	51.9
	UNOPose	✓	1	1	85.9	77.4	72.3	65.7	67.3	63.8
	CoG	✓	1	1	57.4	50.2	55.1	48.3	75.2	69.3
	Ours	✓	1	1	**88.9**	**77.6**	**81.1**	**71.7**	**79.3**	**74.3**
UNOPose (coarse)	No refine	–	–	–	65.8	54.9	64.3	53.5	53.8	47.9
	FoundationPose	✗	16	5	77.1	72.1	76.1	72.0	84.3	75.7
	ICP-Recon	✗	16	$\leq$30	69.4	58.8	57.4	49.8	77.0	73.4
	Ours	✓	16	5	**78.9**	**75.0**	**83.6**	**77.7**	**88.0**	**79.0**
	FoundationPose-1st	✗	1	5	67.6	59.7	71.0	62.7	29.8	31.8
	ICP-Depth	✓	1	$\leq$30	63.3	52.4	57.1	48.1	69.2	62.0
	UNOPose	✓	1	1	73.3	63.8	69.3	62.7	75.3	71.4
	CoG	✓	1	1	59.9	50.1	60.1	52.0	83.8	76.7
	Ours	✓	1	1	**75.1**	**65.6**	**75.2**	**67.9**	**86.5**	**81.0**
CoG (coarse)	No refine	–	–	–	45.8	36.8	54.7	43.0	43.2	41.9
	FoundationPose	✗	16	5	57.5	51.9	65.1	62.8	82.6	74.9
	ICP-Recon	✗	16	$\leq$30	49.8	39.1	49.0	41.9	74.5	70.6
	Ours	✓	16	5	**64.8**	**61.8**	**83.6**	**77.7**	**85.5**	**77.4**
	FoundationPose-1st	✗	1	5	54.5	45.5	62.8	56.4	54.3	49.0
	ICP-Depth	✓	1	$\leq$30	48.2	38.1	51.0	43.6	67.2	60.6
	UNOPose	✓	1	1	54.0	45.3	64.2	57.3	72.7	68.9
	CoG	✓	1	1	51.3	41.7	59.6	50.3	80.3	73.9
	Ours	✓	1	1	**56.9**	**48.3**	**65.0**	**60.4**	**81.8**	**77.1**

For multi-reference refinement, the baseline methods include FoundationPose (Wen et al., 2024) and ICP. To refine the pose using FoundationPose, we reconstruct the 3D meshes with 16 randomly-selected reference views. The vertices of the reconstructed 3D meshes are leveraged as the reference point clouds by ICP, a variant that we refer to as ICP-Recon. Compared with single-view ICP, which we refer to as ICP-Depth, ICP-Recon can provide a complete point cloud for reference.

Influence of Initialization Methods. To demonstrate the robustness to initialization, refiners are evaluated given coarse poses produced by three one-reference methods: CoordAR (Zuo et al., 2026), UNOPose (Liu et al., 2025b) and COG (Che et al., 2026). Both UNOPose and COG employ a two-stage pipeline consisting of coarse pose estimation and subsequent refinement; thus, we use their coarse predictions for initialization and compare our approach against their respective refinement modules. To further assess generalizability across initialization sources, we also include CoordAR, a method without a refinement stage. As presented in Tab. 1, our method consistently outperforms competing refiners across all initializations, demonstrating superior robustness and refinement capacity regardless of the quality of the initial coarse pose.

Qualitative Results. We show qualitative comparisons of 6D object pose estimates in Fig. 4, where all refiners start from coarse poses produced by COG. Across challenging cases involving textureless surfaces (rows 1,2,3) and large viewpoint variations (rows 1,2,4), initial coarse poses exhibit notable misalignment. Under large viewpoint variations, correspondence-based methods such as UNOPose and COG are particularly prone to failure, frequently establishing false correspondence predictions (highlighted in red) that severely degrade final pose estimation. Additionally, FoundationPose suffers from artifacts when reconstructing 3D models from a single reference view. In contrast, our direct regression approach effectively corrects initial alignment errors without establishing explicit correspondences, producing bounding boxes that closely align with the ground truth.

Table 2: Refinement runtime for one-reference novel-object 6D pose refinement. All methods start from the same coarse initial poses and timed over their refinement / second stage only. We report latency, frames per second(FPS) and peak GPU memory at batch size 1.

Refiner	#Ref.	#Iter.	Recon. (s)	FPS ↑	Peak mem. (GB)↓
FoundationPose	16	5	96	**14.2**	**0.2**
Ours	16	1	–	5.0	5.8
UNOPose	1	1	–	**12.3**	**1.5**
COG	1	1	–	10.9	1.0
Ours	1	1	–	**12.3**	1.9

Table 3: Ablation study of the proposed refiner on LINEMOD dataset with single refine iteration on COG coarse poses. Best results in **bold**.

Variant	Component					LINEMOD		YCB-V	
	DT	SD-VAE	Point feat.	Pos. Enc.	Agg.	ADD(-S) ↑	AR ↑	ADD(-S) ↑	AR ↑
Full	✓	✓	✓	4D RPE	Attn.	**56.9**	**48.3**	65.0	**60.4**
A1	✓	✗	✓	4D RPE	Attn.	56.8	48.2	65.0	60.1
B1	✓	✓	✗	4D RPE	Attn.	56.4	47.9	**65.6**	60.2
B2	✓	✓	✗	–	Attn.	42.0	33.4	51.4	39.2
B3	✓	✓	✓	Abs.	Attn.	53.9	44.3	63.9	54.7
C1	✓	✓	✓	4D RPE	Pool.	56.7	47.1	64.7	58.9
D1	✗	✓	✓	4D RPE	Attn.	56.6	47.8	64.4	58.9

Runtime. We evaluate computational efficiency in Tab. 2. In the one-reference setting, our model runs at 12.3 FPS with 1.9 GB peak memory, matching UNOPose (12.3 FPS) and outperforming COG (10.9 FPS). For multi-reference setups, FoundationPose incurs a high 96 s pre-computation overhead for 3D reconstruction, whereas our reconstruction-free framework completely eliminates this cost. With 16 reference views, InstantRefine achieves 5.0 FPS with 5.8 GB memory, delivering efficient, near-real-time refinement without requiring 3D mesh generation.

4.3 Ablation Studies

Network Components. Tab. 3 presents the ablation study of the proposed refiner on LINEMOD and YCB-V, using COG coarse poses and a single refine iteration. We evaluate the contributions of Decoupled Translation (DT), SD-VAE, point features, positional encoding, and aggregation. Compared with the full model (Ours), removing SD-VAE (A1) only causes marginal degradation on LINEMOD but larger degradation in AR on texture-rich YCB-V, indicating that SD-VAE provides limited gains on texture-less objects but auxiliary gains for texture-rich ones. Removing point features (B1) causes a larger performance drop on LINEMOD than on YCB-V, suggesting that point features can provide critical information for texture-less LINEMOD objects. When removing both point features and positional encoding (B2), performance collapses severely across both datasets, underscoring the vital role of explicit 3D spatial representations. Replacing the 4D relative positional encoding (RPE) with absolute positional encoding (B3) also causes clear degradation, validating the superiority of 4D RPE for 3D spatial awareness. Replacing attention with pooling (C1) notably reduces AR on both LINEMOD and YCB-V, showing that attention-based aggregation better fuses multi-source features. Finally, regress entangled translation (D1) also leads to a consistent performance drop, validating the benefit of using decoupled translation targets (detailed formulations are provided in Appendix). Overall, the full model achieves the best or highly competitive performance across all metrics, confirming the effectiveness and joint necessity of each component.

Iterations. To balance performance and efficiency, we default to a single refinement iteration in our main one-reference evaluation; additional iterations can be dynamically applied when higher accuracy is required at the cost of runtime.Fig. 6 illustrates the impact of the number of refinement iterations on LINEMOD when initialized with COG's coarse poses. Both ADD(-S) and AR exhibit steady performance gains as the number of iterations increases, confirming that iterative refinement progressively improves pose accuracy. Notably, the first iteration yields the most substantial im-

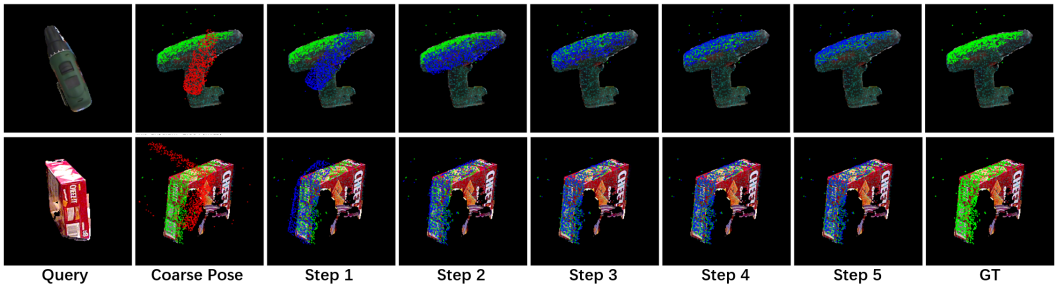

Figure 5: Qualitative visualization of iterative refinement steps. We illustrate how the query points reprojected onto the reference view evolve over successive refinement iterations ($t = 1 \dots 5$). Red, blue, and green denote the initial coarse pose, our refined pose at each step, and the ground-truth pose, respectively.

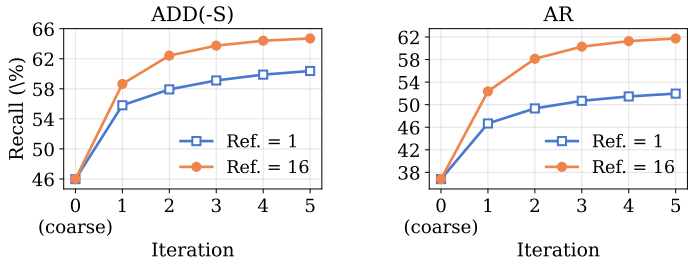

Figure 6: Iterative refinement performance on LINEMOD initialized with COG coarse poses (ADD(-S) left, AR right). Iteration t takes the prediction from Iteration $t-1$ as input, where Iteration 0 represents the initial coarse pose.

provement, accounting for the vast majority of total gains. Fig. 5 shows how poses are updated at each step.

5 Limitations

While InstantRefine achieves strong performance for one-reference pose refinement, it has a few limitations. First, as an RGB-D-based framework, it relies on depth maps sampled from sensor observations, making it currently inapplicable to RGB-only settings or scenarios with severely degraded depth (e.g., highly transparent or specular metallic surfaces). Second, similar to existing refinement architectures, our delta pose regression relies on a reasonable initialization; when presented with extreme initialization errors, the search space for incremental pose updates may become challenging, where integrating coarse pose resampling could serve as a valuable future direction.

6 Conclusion

We present InstantRefine, a simple yet effective plug-and-play framework for one-reference 6D object pose refinement. By formulating pose refinement as direct delta pose regression over point tokens, InstantRefine completely eliminates the need for 3D model reconstruction, explicit descriptor matching, and geometric pose solvers. Furthermore, our architecture naturally extends to multi-view reference settings to incorporate rich spatial context. Extensive experiments across benchmark datasets demonstrate that InstantRefine achieves state-of-the-art accuracy under various pose initializations and extreme viewpoint variations while maintaining lightweight, near-real-time runtime performance.

AI USE STATEMENT

In this work, we used generative AI tools for assist with translation and implement methods. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning and reformatting dataset, supporting qualitative and thematic data analysis, and interpreting results are not applicable to this work. Additionally, we used generative AI tools for creating or editing software code. We have reviewed all AI-assisted work. Specifically, LLMs were used to refine manuscript language for clarity and assist with code implementation. All AI-assisted code was manually tested and validated by the authors, and no synthetic experimental data or visual artifacts were generated. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we provide detailed descriptions of our methodology, architecture, and training objectives in Sec. 3 and Sec. A.1. Comprehensive evaluation setups are further documented in Sec. 4.1 and Sec. A.2.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pp. 586–606. Spie, 1992.

Junhao Cai, Yisheng He, Weihao Yuan, Siyu Zhu, Zilong Dong, Liefeng Bo, and Qifeng Chen. Ov9d: Open-vocabulary category-level 9d object pose and size estimation. *arXiv preprint arXiv:2403.12396*, 2024.

Yuchen Che, Jingtu Wu, Hao Zheng, and Asako Kanezaki. Cog: Confidence-aware optimal geometric correspondence for unsupervised single-reference novel object pose estimation. *arXiv preprint arXiv:2603.00493*, 2026.

Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1724–1734, 2014.

Jaime Corsetti, Davide Boscaini, Changjae Oh, Andrea Cavallaro, and Fabio Poiesi. Open-vocabulary object 6d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18071–18080, 2024.

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*, pp. 13142–13153, 2023.

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)*, pp. 2553–2560. IEEE, 2022.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge, 2010.

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

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*, 35:35103–35115, 2022a.

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*, pp. 6814–6824, 2022b.

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 *2011 international conference on computer vision*, pp. 858–865. IEEE, 2011.

Tomáš Hodaň, Jiří Matas, and Štěpán Obdržálek. On evaluation of 6d object pose estimation. In *European Conference on Computer Vision*, pp. 606–619. Springer, 2016.

Tomáš Hodaň, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, Caner Sahin, Fabian Manhardt, Federico Tombari, Tae-Kyun Kim, Jiří Matas, and Carsten Rother. BOP: Benchmark for 6D object pose estimation. *European Conference on Computer Vision (ECCV)*, 2018.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

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 (CoRL)*, 2022.

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)*, pp. 683–698, 2018.

Lahav Lipson, Zachary Teed, Ankit Goyal, and Jia Deng. Coupled iterative refinement for 6d multi-object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Mengya Liu, Siyuan Li, Ajad Chhatkuli, Prune Truong, Luc Van Gool, and Federico Tombari. One2any: One-reference 6d pose estimation for any object. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6457–6467, 2025a.

Xingyu Liu, Gu Wang, Ruida Zhang, Chenyangguang Zhang, Federico Tombari, and Xiangyang Ji. UNOPose: Unseen object pose estimation with an unposed rgb-d reference image. In *CVPR*, June 2025b.

Xingyu Liu, Ruida Zhang, Chenyangguang Zhang, Gu Wang, Jiwen Tang, Zhigang Li, and Xiangyang Ji. Gdrnpp: A geometry-guided and fully learning-based object pose estimator. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2025c.

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. In *ECCV*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. Gigapose: Fast and robust novel object pose estimation via one correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9903–9913, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4561–4570, 2019.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

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*, pp. 10684–10695, 2022.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6738–6748, 2022.

Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. OnePose: One-shot object pose estimation without CAD models. *CVPR*, 2022.

Martin Sundermeyer, Tomáš Hodaň, Yann Labbe, Gu Wang, Eric Brachmann, Bertram Drost, Carsten Rother, and Jiří Matas. Bop challenge 2022 on detection, segmentation and pose estimation of specific rigid objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2784–2793, 2023.

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*, 30, 2017.

Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. GDR-Net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16611–16621, June 2021.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17868–17879, 2024.

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

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*, pp. 5745–5753, 2019.

Dexin Zuo, Ang Li, Wei Wang, Wenxian Yu, and Danping Zou. Coordar: One-reference 6d pose estimation of novel objects via autoregressive coordinate map generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 14122–14130, 2026.

A APPENDIX

CONTENTS

A.1 Method Details 13
A.2 Evaluation Details 14
A.3 More Experiments Results 14

A.1 METHOD DETAILS

Data Pre-processing. To remove occlusions and background interference, we first apply an object mask to the input RGB images to eliminate all non-object regions. For training data diversity, we do not apply the clean operation during training. To reduce the noise in the depth maps, we use the object mask to extract the corresponding depth regions and further filter out outliers that are far from the object in 3D space. Finally, to obtain the object point cloud, we perform random sampling on the depth map.

Point Cloud Normalization. Both the query and reference point clouds are normalized before being fed into our network. The normalization for each point cloud consists of a centering operation followed by scale normalization. Formally, the normalization transformation is represented by a matrix:

$$\mathbf{N} = \begin{bmatrix} 1/d & 0 & 0 & 0 \\ 0 & 1/d & 0 & 0 \\ 0 & 0 & 1/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{A5}$$

where $[c_x, c_y, c_z]^{\mathrm{T}}$ is the center of the point cloud and d is the diameter of the object in the reference frame, approximated following the approach of CoordAR (Zuo et al., 2026). Specifically, d is derived from both the mask size and the depth:

$$d = \frac{D_{median} \cdot \sqrt{w^2 + h^2}}{(f_x + f_y)/2}, \tag{A6}$$

where D_{median} is the median depth inside the mask, $\{f_x, f_y\}$ are the camera intrinsic focal lengths, and $\{w, h\}$ are the width and height of the visible mask, respectively.

Parametrization. Since the target delta pose $[\Delta\bar{\mathbf{T}}]_{\mathbf{N}}$ is a SE(3) element, in order to regress it by the network, we parametrize it as a 9-dimension vector by a vectorization function $\mathrm{vec}(\cdot)$. For an input $\mathbf{T} \in \mathrm{SE}(3)$ with rotation $\mathbf{R} \in \mathrm{SO}(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$, the function $\mathrm{vec}(\cdot)$ is written as:

$$\mathrm{vec}(\mathbf{T}) \;=\; \left[\, \mathrm{rot6d}(\mathbf{R}) \;\middle|\; \mathbf{t}_{\mathbf{T}^{-1}} \,\right] \;\in\; \mathbb{R}^9, \qquad \mathrm{rot6d}(\mathbf{R}) = [\, \mathbf{R}_{:,0},\ \mathbf{R}_{:,1} \,], \tag{A7}$$

where the $\mathrm{rot6d}(\cdot)$ is the function which convert $\mathbf{R}$ into continuous 6-D representation (the first two columns of the rotation matrix) (Zhou et al., 2019) and the translation slot holds $\mathbf{t}_{\mathbf{T}^{-1}} = -\mathbf{R}^{\top}\mathbf{t}$, i.e. the translation of the *inverse* transform. To understand why we use $\mathbf{t}_{\mathbf{T}^{-1}}$ instead of $\mathbf{t}$, consider the example in Fig. A7, which illustrates the difference. As shown in the figure, our anchor frame is based on the reference frame. Consequently, the translation component of $[\Delta\bar{\mathbf{T}}]_{\mathbf{N}}$ is coupled with the rotation component. In other words, the translation and rotation of $[\Delta\bar{\mathbf{T}}]_{\mathbf{N}}$ both determine the center of the query object in the anchor frame. This kind of coupling can make it difficult for neural networks to learn (Li et al., 2018). However, the translation of $[\Delta\bar{\mathbf{T}}^{-1}]_{\mathbf{N}}$ separately describes the center of query object, thereby avoiding this issue. To recover pose from the parametrization produced by the network, we define a inverse function $\mathrm{vec}^{-1}(\cdot)$ which re-orthonormalizes the rotation by Gram–Schmidt and recomputes the estimated translation $\hat{\mathbf{t}}$ from network output $\hat{\mathbf{t}}_{\mathbf{T}^{-1}}$ by $\hat{\mathbf{t}} = -\mathbf{R}\,\hat{\mathbf{t}}_{\mathbf{T}^{-1}}$.

Training Objective. Unlike correspondence-based methods, which aim to learn descriptors, our method is trained with direct supervision on the pose update. Specifically, we regress the pose update Eq. (4) in the parametrization equation A7, with the rotations and translations weighted independently:

$$\mathcal{L} \;=\; \left\|\mathrm{rot6d}(\hat{\mathbf{R}}) - \mathrm{rot6d}(\bar{\mathbf{R}})\right\|_2^2 \;+\; \lambda \left\|\hat{\mathbf{t}} - \bar{\mathbf{t}}\right\|_2^2, \tag{A8}$$

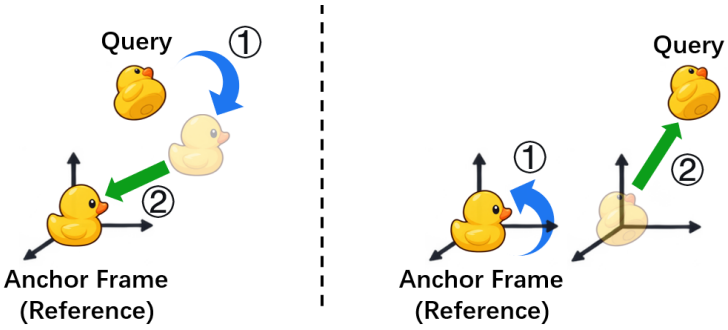

Figure A7: Illustration of translation parametrization. The left figure shows the transformation of $[\Delta\bar{\mathbf{T}}]_{\mathbf{N}}$, where the object are first rotated around the center of the anchor frame and then shifted to the center. The right figure shows the transformation of $[\Delta\bar{\mathbf{T}}^{-1}]_{\mathbf{N}}$, whose translation component solely determines the center of the query object. The circled numbers indicate the order of the transformations.

where λ compensates the different scales of rotation and translation term.

Network Hyperparameters. Tab. A4 presents the hyperparameters of our refiner. As the table shows, we append a view embedding to each view to distinguish the anchor view from the other, as done by VGGT (Wang et al., 2025). We also rescale the normalized point coordinates that parameterize the 4-D rotary embedding by a factor of 13, chosen so that the fastest frequency pair completes one full cycle from the object centre to the edge of the normalized box (i.e. $6.5 \approx 2\pi$).

Training Details. Our models are trained using the AdamW optimizer (Loshchilov & Hutter, 2017) for 400K iterations. The learning rate is warmed up to 1×10^{-4} for the first 20K iterations and then annealed to zero with a cosine schedule after 70% of the training process. Training takes approximately two days on a single NVIDIA RTX 4090 GPU with the batch size set to 16.

A.2 Evaluation Details

Reference Selection. For benchmark evaluations, the choice of reference frames follows standard protocols to ensure fair comparison. For LINEMOD and YCB-V, we consistently adopt the *first frame* of each object sequence as the template reference view, following existing works (Liu et al., 2025a; Zuo et al., 2026). For TUD-Light (TUD-L), where objects exhibit severe lighting changes and larger viewpoint variations, we utilize the *selected* reference frames provided by UNOPose (Liu et al., 2025b). The specific reference frame and segmentation mask settings for both coarse pose estimation and our refiner across datasets are summarized in Tab. A5.

A.3 More Experiments Results

Qualitative Results. We provide qualitative comparisons of refined pose predictions. As illustrated in Fig. A8, we visualize the depth error maps computed by comparing the depth maps rendered from the refined poses against the ground truth depth maps. Our model consistently achieves low depth residual errors across various challenging instances, demonstrating high geometric precision in the regressed pose updates. Furthermore, we visualize the pose refined by our method using multiview reference in Fig. A11.

Robustness to Predicted Masks. To assess the impact of mask quality on point sampling, we evaluate performance when replacing ground truth (GT) masks with SAM-predicted masks (Kirillov et al., 2023). As shown in Tab. A6, switching to predicted masks leads to only minor performance drops across LINEMOD and YCB-V for both UNOPose and our method. Notably, InstantRefine consistently maintains its performance advantage over UNOPose under predicted masks, demonstrating that our spatial point tokenization is robust to imperfect object segmentations in real-world pipelines.

Sensitivity to Depth Noise. To evaluate robustness against noisy depth, we inject i.i.d. Gaussian noise $\mathcal{N}(0, \sigma^2)$ into the observed depth maps on LINEMOD, sweeping σ using COG coarse poses. As shown in Fig. A9, both InstantRefine and UNOPose maintain stable accuracy under mild noise

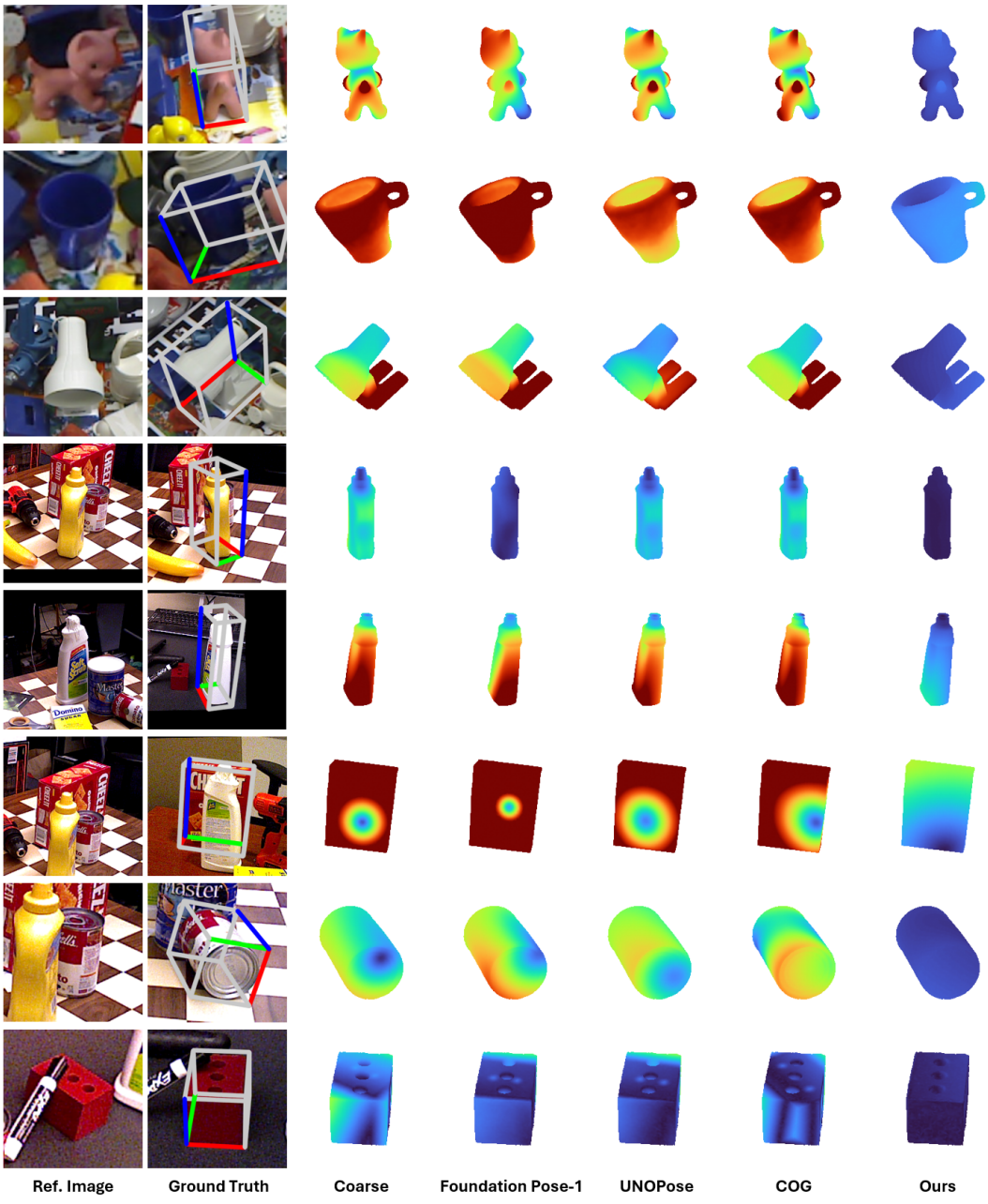

Figure A8: Qualitative visualization of depth error maps. The error is measured as the absolute distance between the ground truth depth map and the depth map rendered from refined pose (legend: 0 cm 5 cm).

Table A4: Implementation details of the InstantRefine, shared by every variant in Tab. 3. Quantities marked *learned* are trained; everything else is fixed by the recipe. The single-reference variant differs only in the two entries marked †: it is trained with $V = 2$.

Backbone	
RGB features	DINOv2-B/14 patch tokens (768-d) and SD-VAE latent (4-d), resized to 28×28 and concatenated (772-d)
Input crop	224×224
Raw points / view	2048 back-projected depth samples
Point feature	LRF-aligned PointNet, radii 0.1 / 0.2 and 64 / 256 neighbours, shared MLP $\rightarrow$ 256-d
Observation tokens	
Tokens / view	$S = 128$ (farthest-point sampling) plus [ROT, TRANS], i.e. 130
Token width	$D = 768$ ($1028 \rightarrow 768$ linear fusion)
Task tokens	one learned [ROT, TRANS] pair, shared by all views and objects (learned)
View embedding	2 entries, anchor / other (learned)
Transformer	
Depth / heads	$L = 12$ blocks, 12 heads, head dim 64
Attention	intra-view and cross-view over the packed sequence, varlen (FlashAttention), rotary-localized
Position encoding	4D RoPE on $[\text{role}, 13\tilde{x}, 13\tilde{y}, 13\tilde{z}]$, 8 frequency pairs per channel $(10000^{-k/8})$
Feed-forward	GeGLU
Read-out	MLP $768 \rightarrow 768 \rightarrow 384 \rightarrow \{6, 3\}$, SiLU
Training	
Init-pose noise	rotation $\mathcal{N}(0, 15^\circ)$ per axis; translation $\mathcal{N}(0, 0.2)$ per axis, in diameter units
Loss	MSE on $\text{vec}(\Delta)$, $\lambda_{\text{trans}} = 10$
Optimizer	AdamW, lr 10^{-4}, weight decay 10^{-5}, gradient norm clipped at 10
Schedule	1% linear warmup, flat, cosine decay after 70%, to 0
Batch / precision	16 samples (6 in the multi-view recipe), bf16 mixed precision, seed 12345
Epochs / steps	20 epochs, 20k iterations per epoch cap
Training data	FoundationPose-GSO, FoundationPose-Objaverse and OV9D, re-rendered with PASCAL VOC (Everingham et al., 2010) background and colour augmentation
Views per sample	$V \sim \mathcal{U}\{2, \ldots, 9\}$, $K = V - 1$ references †

Table A5: Evaluation protocol setup regarding reference frames and masks for coarse pose estimators and our refiner across datasets.

Dataset	Coarse Pose		Refiner	
	Reference	Mask	Reference	Mask
LINEMOD	First	GT	First	GT
YCB-V	First	GT	First	GT
TUD-L	Selected	Predicted	Selected	Predicted

levels. However, as depth noise escalates, UNOPose experiences a sharp ADD(-S) degradation, whereas our method exhibits a significantly smoother decline and stabilizes at a higher baseline. Consequently, the performance margin of InstantRefine over UNOPose steadily widens with increasing noise, demonstrating superior resilience to sensor depth noise.

Influence of Initialization. To investigate the impact of coarse pose initialization on our refiner, we evaluate the pose refinement performance on LINEMOD using COG as the initial coarse pose baseline. As illustrated in Fig. A10, similar to most pose refinement methods, our method exhibits a characteristic rotation error threshold beyond which rotation optimization becomes increasingly

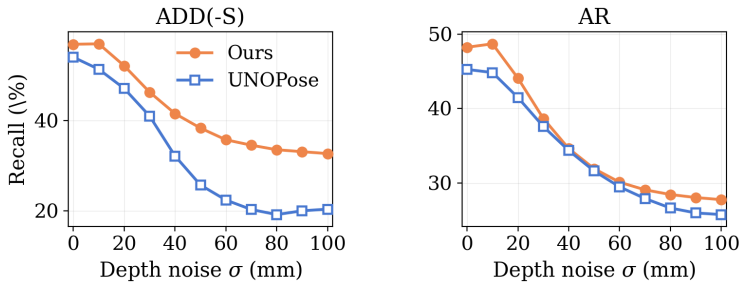

Figure A9: Sensitivity to depth noise on LINEMOD initialized with COG coarse poses. Accuracy is measured under varying levels of zero-mean Gaussian depth noise $\mathcal{N}(0, \sigma^2)$. InstantRefine exhibits a noticeably smoother degradation and maintains higher recall compared to UNOPose as noise escalates.

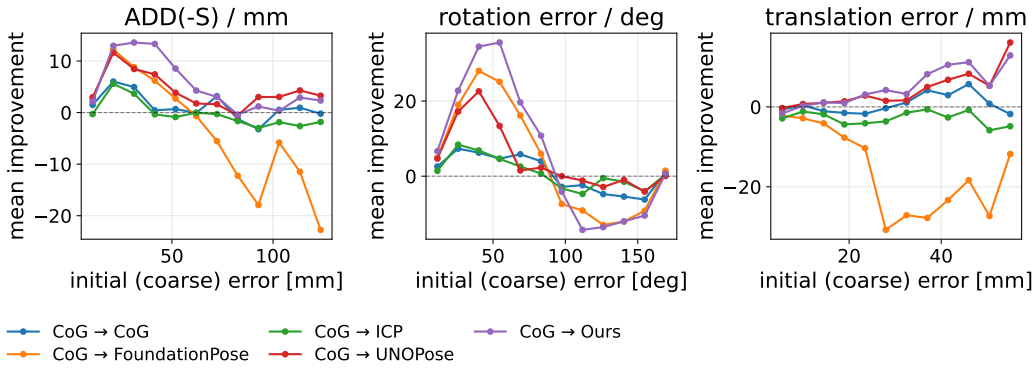

Figure A10: Impact of coarse pose initialization quality on LINEMOD. Mean improvements in ADD(-S), rotation error, and translation error under varying initial coarse pose error bins.

challenging. Crucially, InstantRefine demonstrates a substantial and positive pose optimization capability for coarse initializations with initial rotational errors within $80°$. Furthermore, InstantRefine is remarkably robust to spatial displacements, showing even more pronounced refinement gains when handling instances with larger initial translation errors.

Table A6: Effect of the segmentation mask used to sample the observed 3D points. *GT* means the dataset's ground truth segmentation mask; *SAM* denotes mask produced from the ground truth bounding box by SAM (Kirillov et al., 2023) ViT-H model.

Mask	Method	LINEMOD		YCB-Video	
		ADD(-S) ↑	AR ↑	ADD(-S) ↑	AR ↑
GT	UNOPose	54.0	45.3	64.2	57.3
	Ours	**56.9**	**48.3**	**65.0**	**60.4**
SAM	UNOPose	53.8 (-0.2)	44.2 (-1.1)	62.5 (-1.7)	55.3 (-2.0)
	Ours	**56.5** (-0.4)	**47.1** (-1.2)	**63.8** (-1.2)	**56.9** (-3.5)

Failure Cases. Although InstantRefine exhibits robust refinement capabilities across various challenging scenarios, certain extreme pose initializations still lead to failure, as visualized in Fig. A12.

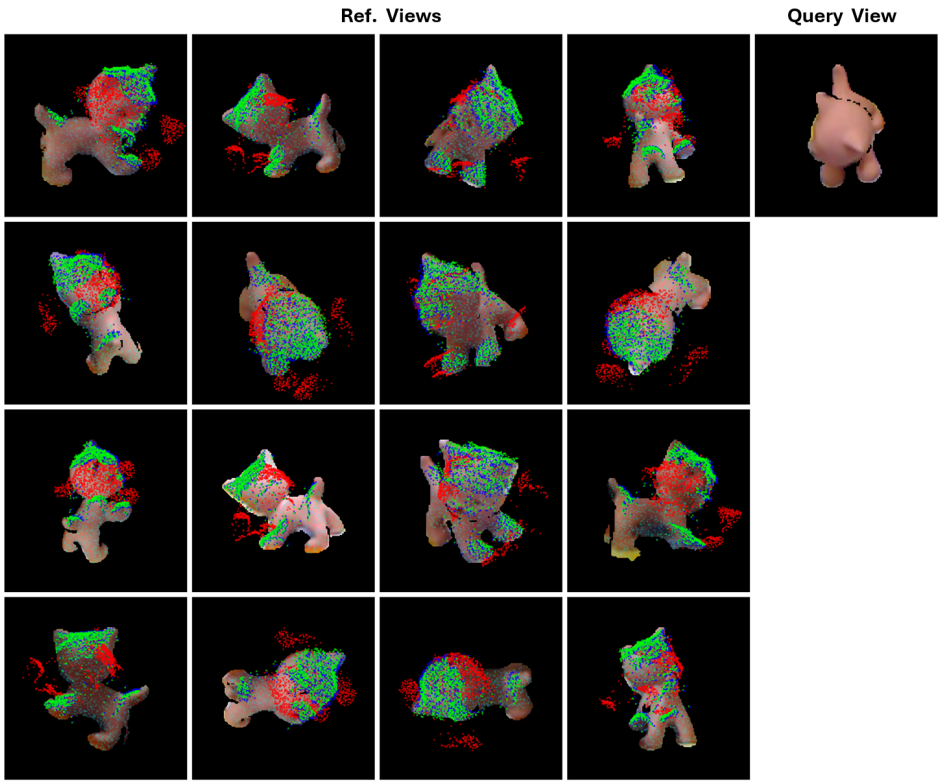

Figure A11: Qualitative illustration of pose refinement using multiview references. Points from the query view are reprojected onto the reference view using the coarse, ground-truth, and refined poses. Red, green, and blue denote the coarse pose, the ground-truth pose, and the pose refined by our method, respectively.

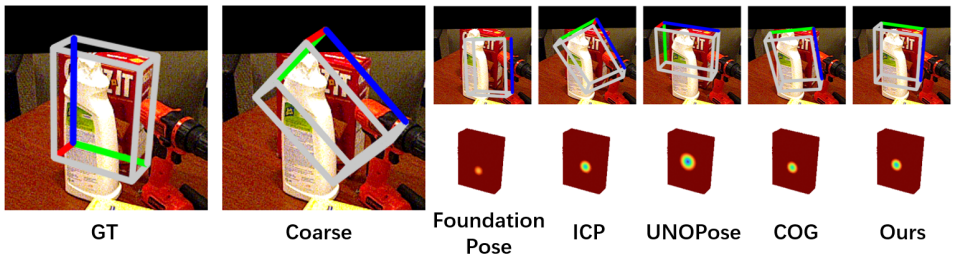

Figure A12: Failure Cases. The top-right row shows the refined pose, while the bottom-right row visualize the depth error with the refined pose (legend: 0 cm 5 cm).