POSEVLM: ADAPTING VISUAL-LANGUAGE MODEL FOR 6DOF OBJECT POSE ESTIMATION

Anonymous authors
Paper under double-blind review

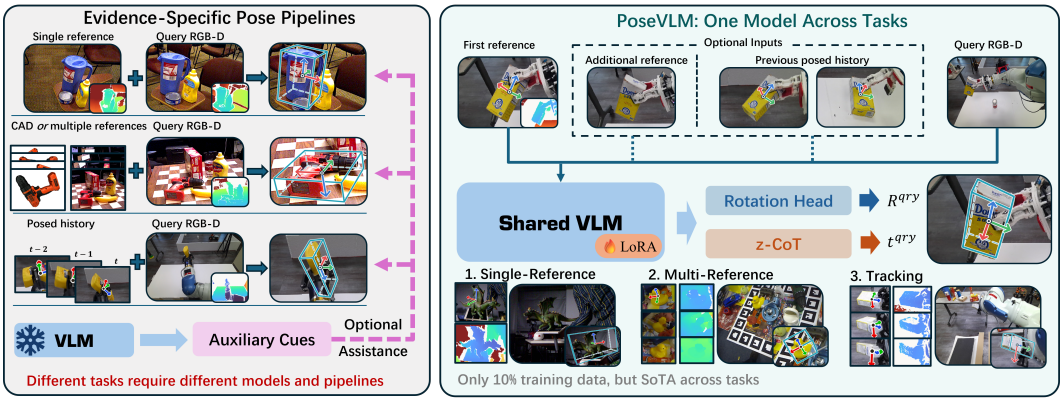

Figure 1: **PoseVLM unifies 6DoF pose estimation across single references, multiple references, and tracking.** Existing methods couple single-reference observations, CAD or multi-reference evidence, and tracking history to different pose pipelines or instance-specific reconstruction (left). PoseVLM instead expresses these settings through one common input structure (right): a posed first reference defines the object frame, optional additional references or posed history provide extra evidence, and an unannotated query RGB-D frame is the prediction target. A shared VLM feeds a dedicated rotation head and a geometry-grounded z-axis chain-of-thought (z-CoT) translation branch to recover the query pose. The lower examples show the same model applied to different tasks.

ABSTRACT

6DoF object pose estimation appears in three settings that share one objective but differ in the available evidence: a single reference image, multiple references, or tracking. Existing methods require task-specific pipelines or instance-specific reconstruction, while VLM-based methods use VLMs only to generate auxiliary cues for conventional pose estimators. We introduce PoseVLM, a unified VLM that directly predicts metric 6DoF pose across these settings with a single checkpoint. Our diagnostics show that language decoding learns translation readily but struggles with rotation. PoseVLM thus retains language-based translation and uses a coarse-to-fine rotation head that accommodates valid alternatives from neighboring bins and object symmetry. Since observed depth measures surfaces rather than object centers, geometry-grounded z-axis chain-of-thought recovers center depth with a cross-view-inferred signed offset. With 10% of One2Any's pose-training images, PoseVLM outperforms representative baselines across all three settings without test-time CAD models or instance reconstruction.

1 INTRODUCTION

Estimating an object's 3D rotation and translation (6DoF pose) is a core capability for robotic manipulation (Kappler et al., 2018; Wen et al., 2022; Yoshida et al., 2025), augmented reality (Amara et al., 2024), and scene understanding. The task now spans known and novel objects specified by CAD geometry or reference observations (Labbé et al., 2022; Wen et al., 2024), as well as single-frame

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.

estimation and tracking, but shares one goal: recover the metric pose in a coordinate frame defined by the available reference evidence, rather than by recognizing a specific training object.

Existing solutions, however, remain tightly coupled to particular forms of object evidence. Instance- and category-level networks require target-specific training or predefined class priors, limiting their transfer to arbitrary novel objects (Xiang et al., 2017; Su et al., 2022; Cai et al., 2024; Zhang et al., 2023a). CAD- and multi-reference methods improve novel-object generalization, but typically rely on a CAD model, pose initialization, rendering, retrieval, or iterative alignment (Labbé et al., 2022; Wen et al., 2024). Single-reference methods reduce these requirements, yet resolve ambiguities from occlusion, symmetry, and viewpoint changes through specialized correspondences, priors, or refinement (Fan et al., 2024; Liu et al., 2025a; Zuo et al., 2026; Liu et al., 2026; Luo et al., 2026); tracking adds dedicated pose propagation, mapping, or online reconstruction and risks accumulated drift (Liu et al., 2024a;b; Wen et al., 2023). Consequently, Figure 1 (left) shows that moving from one reference to multiple observations or posed history (previous frames with known or predicted object poses) typically changes the pose pipeline or requires reconstructing a model, rather than simply changing the evidence supplied to the same predictor. A unified predictor can incorporate newly available views or tracking history as additional evidence, without rebuilding an instance model or switching pose pipelines.

VLMs offer pretrained visual knowledge, multi-observation integration, and reasoning that promise broad generalization from limited pose-specific data (Achiam et al., 2023; Bai et al., 2025); geometry-oriented variants already demonstrate metric spatial reasoning and depth prediction (Chen et al., 2024a; Cheng et al., 2024; Cai et al., 2025; Yu et al., 2026). Yet they have not established direct object-centric $\mathrm{SE}(3)$ estimation: Figure 1 (left bottom) shows that existing VLM-based systems use language for identification, ambiguity handling, or candidate selection, while specialized estimators produce the metric pose (Guo et al., 2025; Liu et al., 2025b; Ding et al., 2024). This leaves open how a VLM should represent and decode a complete 6DoF pose from RGB-D observations.

We therefore develop **PoseVLM**, a unified model with VLM-adaptation designs guided by diagnostic experiments. As Figure 1 (right) illustrates, PoseVLM unifies a full-frame query with all available evidence: the first posed reference establishes the canonical frame, while optionally additional observations or posed history provide complementary information. Our diagnostics reveal a decoding asymmetry: language generation learns translation readily but struggles with continuous rotation, while visual features support coarse orientation selection and local refinement. Rotation also presents a supervision-geometry mismatch: single-target supervision discards valid alternatives from neighboring bins and object symmetry. We therefore employ **dual branch** designs, retaining language-based translation and using a multi-target coarse-to-fine rotation head. Translation presents a surface-center observation gap: surface depth does not directly reveal canonical-center depth. **Geometry-grounded z-axis chain-of-thought (z-CoT)** bridges this gap by combining observed surface depth with a cross-view-inferred signed surface-to-center offset. Together, these components support single-reference, multi-reference, and tracking tasks with one VLM (Figure 1, right). Our contributions are as follows.

- We introduce **PoseVLM**, a unified VLM that directly estimates metric 6DoF pose from single references, multiple references, or posed history, consuming all evidence directly without per-instance reconstruction or task-specific heads.
- We reveal a mismatch between autoregressive decoding and multi-target rotation supervision, and address it with a **dual-branch** decoder combining language-based translation with multi-solution-aware coarse-to-fine rotation decoding.
- We identify object center depth as the main translation bottleneck and introduce geometry-grounded **z-CoT**, which recovers it from visible surface depth and cross-view geometry.

2 Related Works

Object pose estimation across reference settings. Pose estimation progressed from instance-specific regression and correspondence models (Xiang et al., 2017; Peng et al., 2019; Wang et al., 2021; Su et al., 2022) to category-level and generative models using shared canonical, shape, or semantic priors (Wang et al., 2019; Chen et al., 2021; Di et al., 2022; Cai et al., 2024; Zhang et al., 2023a), but remains tied to known instances or predefined categories. Novel-object methods instead

provide object knowledge at test time through CAD rendering and refinement or posed multi-view and SfM representations (Li et al., 2018; Labbé et al., 2022; Örnek et al., 2024; Lin et al., 2024; Wen et al., 2024; Liu et al., 2022; Sun et al., 2022). Single-reference methods reduce onboarding further, but retain pairwise pipelines involving segmentation, correspondence or coordinate prediction, iterative alignment, explicit priors, and geometric solvers (Fan et al., 2024; Liu et al., 2025a; Zuo et al., 2026; Liu et al., 2026; Luo et al., 2026). Consequently, these reference settings remain organized around pose pipelines tailored to their evidence.

Object pose tracking. Tracking methods filter propagated poses, maintain RGB-D pose graphs or online reconstructions, or map monocular video to proxy geometry (Wen et al., 2021; Deng et al., 2021; Wen & Bekris, 2021; Wen et al., 2023; Zhang et al., 2026). They require dedicated tracking machinery; most recently, ProxyPose (Zhang et al., 2026) recasts tracking as video-to-video translation, following a marked surface region through a synthetic proxy. Tracking is thereby formulated as motion estimation over the video itself, rather than as object-pose prediction based on reference evidence, leaving tracking separate from reference-based estimation.

VLM reasoning for geometric perception. Modern VLMs provide broad visual reasoning (Bai et al., 2025; Achiam et al., 2023), but geometric adaptations focus on spatial relations or metric depth rather than object-centric $\mathrm{SE}(3)$ prediction (Chen et al., 2024a; Cheng et al., 2024; Cai et al., 2025; Yu et al., 2026). Pose systems still use language or VLM features within specialized pipelines for cross-view registration, viewpoint or coordinate prediction, candidate selection, semantic constraints, and ambiguity scoring (Corsetti et al., 2024; 2025; Chen et al., 2024b; Ding et al., 2024; Guo et al., 2025; Liu et al., 2025b). Generic chain-of-thought (CoT) methods improve answer-oriented reasoning (Wei et al., 2022; Zhang et al., 2023b; Mitra et al., 2024), but are not designed for metric center-depth reasoning or multi-target $\mathrm{SO}(3)$ supervision. PoseVLM instead turns VLM reasoning into direct metric pose prediction by combining geometry-grounded center-depth reasoning with a jointly trained head for multi-target-aware coarse-to-fine rotation decoding.

3 METHOD

Given one or more reference RGB-D observations and a query RGB-D observation, PoseVLM estimates the object's rotation $\widehat{\mathbf{R}}^{\mathrm{qry}}$ and translation $\widehat{\mathbf{t}}^{\mathrm{qry}}$ in the query camera frame. Throughout this section, a hat denotes a prediction and $*$ denotes a ground-truth target. As illustrated in Figure 2, PoseVLM uses a common formulation across single-reference estimation, multi-reference estimation, and tracking, and predicts pose through coarse-to-fine rotation decoding and geometry-grounded z-CoT for translation decoding.

3.1 PROBLEM FORMULATION AND DUAL-BRANCH MOTIVATION

Unified problem formulation. Let $\mathcal{X}^{\mathrm{ref}} = \{\mathbf{X}_i^{\mathrm{ref}}\}_{i=1}^{N}$ denote N reference observations of a target object, where $\mathbf{X}_i^{\mathrm{ref}} = (\mathbf{I}_i^{\mathrm{ref}}, \mathbf{D}_i^{\mathrm{ref}}, \mathbf{K}_i^{\mathrm{ref}})$ contains an RGB image, its aligned metric depth map, and the camera intrinsics; the query observation $\mathbf{X}^{\mathrm{qry}}$ is defined analogously. Each RGB-D observation is encoded as an ordered pair of RGB and depth maps rendered as color images with a fixed color-to-depth scale. Since the pose of a novel object is meaningful only relative to an object-centric coordinate frame, the first posed reference defines this frame. We write $\mathbf{T}_i^{\mathrm{ref}} = [\mathbf{R}_i^{\mathrm{ref}} \mid \mathbf{t}_i^{\mathrm{ref}}]$ and $\mathbf{T}^{\mathrm{qry}}$ for object poses that map canonical object coordinates into the corresponding camera frames. The rotation and projected object center of the first reference are rendered as a coordinate triad on its RGB image, while its center depth and camera intrinsics are included in the prompt to retain the metric information unavailable from the 2D overlay. Given the references and a full-frame query, PoseVLM predicts $\widehat{\mathbf{T}}^{\mathrm{qry}} = [\widehat{\mathbf{R}}^{\mathrm{qry}} \mid \widehat{\mathbf{t}}^{\mathrm{qry}}]$.

Dual-branch motivation. Under the experiment protocols in Appendix E, we first test whether a VLM can generate the complete 6DoF pose as text, but direct fine-tuning achieves only 2.6% ADD-$0.1d$. To identify the source of this failure, we predict each component separately while setting the other to ground truth. The language decoder reaches 80.4% when predicting translation but only 3.0% when predicting rotation, showing that rotation is the main bottleneck. This finding motivates us to introduce a dedicated head for rotation. We then ask whether translation should also be predicted through a dedicated head. However, the translation-head variant reaches only 54.6%, substantially below the 80.4% achieved by the language decoder. We therefore adopt a dual-branch design that

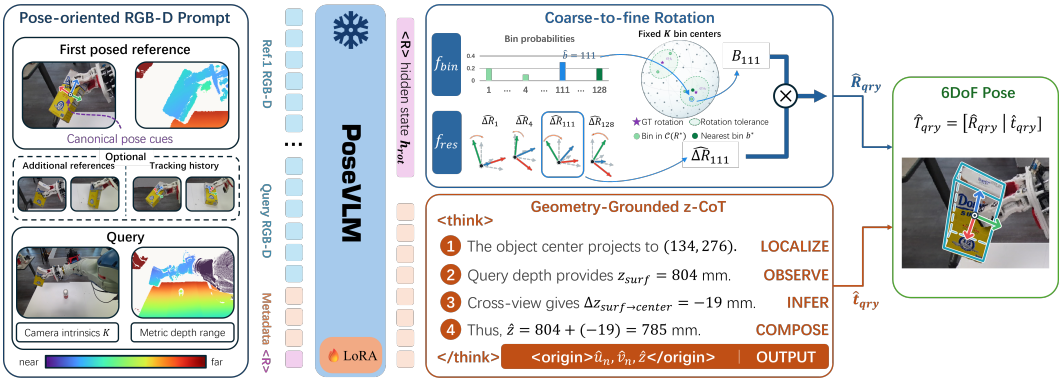

Figure 2: **Overview of PoseVLM.** A frozen VLM adapted with LoRA encodes a pose-oriented RGB-D prompt containing reference observations and a full-frame query. The <R> state recovers rotation through coarse-bin selection and bin-conditioned residual refinement, covering alternative targets caused by neighboring bins and object symmetry. The language decoder recovers translation through geometry-grounded z-CoT, combining observed surface depth with a cross-view-inferred surface-to-center offset. In the lower branch, the yellow surface marker and purple center marker nearly coincide in the query view but separate in the posed reference. The resulting $\widehat{\mathbf{R}}^{\mathrm{qry}}$ and back-projected $\widehat{\mathbf{t}}^{\mathrm{qry}}$ form the final 6DoF pose $\widehat{\mathbf{T}}^{\mathrm{qry}}$.

uses a dedicated head for rotation while retaining the language decoder for translation. Appendix E provides the complete protocols, comparisons, and results.

3.2 Coarse-to-Fine Rotation Decoding

Following the dual-branch design, we first test whether a dedicated head can directly predict rotation. It achieves 52.7% ADD-$0.1d$, over the single language branch, but still falls short of reliable pose estimation. We therefore explore a rotation prediction formulation that better matches the VLM's strengths. Our tests show that the VLM can accurately select a coarse orientation, reaching 92.3% selection accuracy among 32 candidates, and that predicting a rotation near the target is easier than unrestricted prediction, improving rotation-only ADD-$0.1d$ from 3.0% to 12.9% via text generation. These findings motivate a coarse-to-fine rotation decoder, which first selects a coarse rotation from a fixed set of rotation-bins and then predicts a local residual relative to the selected rotation.

Coarse-to-fine rotation factorization. We represent the rotation space with a fixed set of K rotation-bin centers, $\mathcal{B} = \{\mathbf{B}_b\}_{b=1}^{K} \subset \mathrm{SO}(3)$. They are selected via farthest-point sampling under the $\mathrm{SO}(3)$ geodesic distance, which spreads them across the rotation space, and shared for all objects, datasets, and tasks. For each bin b, the model predicts a residual $\widehat{\Delta\mathbf{R}}_b$ and combines it with the bin center: $\widehat{\mathbf{R}}_b = \mathbf{B}_b\widehat{\Delta\mathbf{R}}_b$. Thus, rotation is factorized into coarse-bin selection and bin-conditioned residual refinement within one forward pass.

Multiple coarse rotation targets. Discretizing rotation space can leave several neighboring bin centers similarly close to a target, each requiring only a small residual. Supervising only the nearest bin therefore discards useful alternatives and causes over-fitting. We instead supervise a coarse target set $\mathcal{C}(\mathbf{R}^*)$ which also accounts for object symmetry, since different rotations may represent the same physical pose. We denote the object's symmetry transformations, including the identity, by $\mathcal{S}$. For asymmetric objects, $\mathcal{S} = \{\mathbf{I}_3\}$; continuous symmetries are discretized at a fixed angular interval. For each symmetry-equivalent target $\mathbf{R}^*\mathbf{S}$, we measure the bin distance $d_{b,\mathbf{S}} = d_{\mathrm{SO}(3)}(\mathbf{B}_b, \mathbf{R}^*\mathbf{S})$ and accept all bins within the angular tolerance δ of each

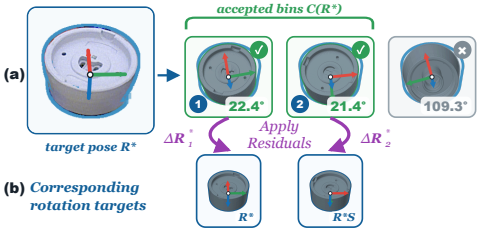

Figure 3: **Multiple correct rotation targets.** (a) Supervision accepts $\mathcal{C}(R^*)$, including bins near the target rotation and bins near the symmetry-equivalent rotation, while rejecting distant ones; (b) each accepted bin can be refined to its own target via its own residual.

target's nearest bin: $\mathcal{C}(\mathbf{R}^*)=\bigcup_{\mathbf{S}\in\mathcal{S}}\{b \mid d_{b,\mathbf{S}} \le \min_{b'} d_{b',\mathbf{S}}+\delta\}$. As Figure 3(a) shows, the union covers bins near the target rotation itself and bins near each symmetry-equivalent rotation. For obtaining residual target of each $b \in \mathcal{C}(\mathbf{R}^*)$, we select the symmetry-equivalent target closest to that bin center and compute

$$\mathbf{S}_b^* = \arg\min_{\mathbf{S}\in\mathcal{S}} d_{\mathrm{SO}(3)}(\mathbf{B}_b, \mathbf{R}^*\mathbf{S}), \quad \mathbf{R}_b^* = \mathbf{R}^*\mathbf{S}_b^*, \quad \Delta\mathbf{R}_b^* = \mathbf{B}_b^{-1}\mathbf{R}_b^*. \tag{1}$$

The corresponding target rotation is $\mathbf{R}_b^* = \mathbf{B}_b\Delta\mathbf{R}_b^*$, as illustrated in Figure 3(b).

Rotation prediction. With the coarse targets and their residuals defined, the rotation head must predict a score and a residual for every bin. We append a trainable `<R>` token before the generated response and use its final-layer hidden state $\mathbf{h}_{\mathrm{rot}}$, which summarizes the visual observations and pose prompt. Then, two lightweight projections map $\mathbf{h}_{\mathrm{rot}}$ to the coarse-bin probabilities $\widehat{\mathbf{p}} = \mathrm{softmax}(f_{\mathrm{bin}}(\mathbf{h}_{\mathrm{rot}})) \in \mathbb{R}^K$ and K bin-specific 6D residuals $\widehat{\mathbf{A}} = \mathrm{reshape}(f_{\mathrm{res}}(\mathbf{h}_{\mathrm{rot}})) \in \mathbb{R}^{K\times 6}$. For bin b, we convert $\widehat{\mathbf{A}}_b$ into $\widehat{\Delta\mathbf{R}}_b \in \mathrm{SO}(3)$ using the continuous 6D rotation representation and orthogonalization (Zhou et al., 2019), and combine it with $\mathbf{B}_b$ to obtain $\widehat{\mathbf{R}}_b$. The rotation head therefore scores and refines all coarse rotations in one forward pass.

Multi-target rotation supervision. We train coarse selection and residual refinement with separate losses. The coarse loss assigns probability to the coarse target set, while the residual loss refines each target bin toward its corresponding rotation:

$$\mathcal{L}_{\mathrm{coarse}} = -\log\left(\sum_{b\in\mathcal{C}(\mathbf{R}^*)} \widehat{p}_b\right), \qquad \mathcal{L}_{\mathrm{res}} = \frac{1}{|\mathcal{C}(\mathbf{R}^*)|}\sum_{b\in\mathcal{C}(\mathbf{R}^*)} d_{\mathrm{SO}(3)}\left(\mathbf{B}_b\widehat{\Delta\mathbf{R}}_b, \mathbf{R}_b^*\right). \tag{2}$$

With one target bin, $\mathcal{L}_{\mathrm{coarse}}$ becomes standard cross-entropy. With multiple targets, the model may concentrate its probability on any single accepted target instead of splitting it among them, while $\mathcal{L}_{\mathrm{res}}$ keeps each accepted bin tied to its own refined target. Overall, the rotation branch scores coarse orientations and refines each with its own residual, while accommodating neighboring and symmetry-equivalent targets.

3.3 Geometry-Grounded z-CoT for Translation

After improving rotation, ADD-$0.1d$ achieves 75.9%, but is still unreliable. We next locate the bottleneck of pose accuracy by replacing single components with ground truth: replacing only t_z raises ADD-$0.1d$ from 75.9% to 96.2%, whereas replacing t_x and t_y together reaches 91.1%. This result shows the object center depth t_z is the dominant source of translation error. Predicting t_z is difficult because an RGB-D observation measures the visible object surface rather than the object center, which often lies inside the object. Near the projected center (u, v) we select a visible surface point P on the object, whose depth is z_{surf}; the canonical center O has depth t_z. Their depths satisfy $t_z = z_{\mathrm{surf}} + \Delta z_{\mathrm{surf}\to\mathrm{center}}$, where z_{surf} can be directly observed from query depth, while the signed camera-z offset $\Delta z_{\mathrm{surf}\to\mathrm{center}}$ must be inferred from cross-view geometry: in the query view P and O project to nearly the same pixel, so their separation is hidden along the viewing ray, whereas the posed reference observes the object from a different direction and makes the offset measurable. We therefore guide the language decoder to recover these two terms via chain of thought.

Language decoding and backprojection. The language decoder ends its z-CoT with `<origin>` $\widehat{u}_n, \widehat{v}_n, \widehat{z}$ `</origin>`, where $(\widehat{u}_n, \widehat{v}_n) \in [0, 1000]^2$ is the normalized canonical-center projection and $\widehat{z}$ is its depth in millimeters. We retain the coordinate format of the pretrained VLM for the 2D coordinates and train PoseVLM's language decoder to predict metric depth directly, motivated by prior work that VLM can learn metric depth (Cai et al., 2025). For an image of size (W, H), we recover $\widehat{u} = W\widehat{u}_n/1000$ and $\widehat{v} = H\widehat{v}_n/1000$, then backproject to 3D with the query intrinsics:

$$\widehat{\mathbf{t}}^{\mathrm{qry}} = \widehat{z}(\mathbf{K}^{\mathrm{qry}})^{-1}\begin{bmatrix}\widehat{u} & \widehat{v} & 1\end{bmatrix}^\top. \tag{3}$$

Geometry-grounded z-CoT reasoning. The lower branch of Figure 2 shows z-CoT reasoning in four steps and Figure 4 illustrates the process: (1) locate the canonical-center point (u, v); (2) read the surface depth z_{surf} near the projected center; (3) infer $\Delta z_{\mathrm{surf}\to\mathrm{center}}$; (4) compose t_z and emit `<origin>` for backprojection. These intermediate targets are constructed offline from ground-truth poses, depth, and object meshes. At inference, PoseVLM uses only the RGB-D observations and the posed reference, without CAD models or any others. Appendix H details target construction.



Figure 4: **z-CoT steps.** Yellow marks the visible surface point P; purple marks the object center O, which lies inside the object. (1) and (2) show the query RGB and depth around the projected center (u, v); (3) shows the posed reference, whose different known pose separates P and O. (4) composes the center depth $t_z = z_{\text{surf}} + \Delta z_{\text{surf}\to\text{center}}$ along the depth axis (yellow: observed; purple: inferred).

3.4 Unified Training and Inference

Mixed-setting training. All three settings share one single-stage training run. Single-reference examples contain one posed RGB-D pair, multi-reference examples add unposed pairs, and tracking examples use posed history with added noise, simulating accumulated drift. Queries remain uncued full frames; reference observations are augmented as full frames or object crops.

Joint optimization. We freeze the pretrained VLM backbone and jointly optimize its LoRA adapters, the newly added <R> embedding, and the rotation head with

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{coarse}}\mathcal{L}_{\text{coarse}} + \lambda_{\text{res}}\mathcal{L}_{\text{res}}, \tag{4}$$

where $\mathcal{L}_{\text{LM}}$ is the loss for z-CoT and the translation result.

Unified inference. The language branch uses greedy decoding, and the final valid <origin> span is parsed and back-projected with Equation 3. The rotation branch selects the maximum-probability bin and its associated residual,

$$\widehat{b} = \arg\max_b \widehat{p}_b, \qquad \widehat{\mathbf{R}}^{\text{qry}} = \mathbf{B}_{\widehat{b}}\widehat{\Delta\mathbf{R}}_{\widehat{b}}. \tag{5}$$

The two outputs form $\widehat{\mathbf{T}}^{\text{qry}} = [\widehat{\mathbf{R}}^{\text{qry}} \mid \widehat{\mathbf{t}}^{\text{qry}}]$. During tracking, this estimate becomes the next frame's pose prompt, enabling frame-by-frame inference without a separate reconstruction module, detector, or tracking-specific head.

4 Experiments

4.1 Experimental Setup

Implementation Details. PoseVLM freezes Qwen3-VL-8B-Instruct (Bai et al., 2025) and trains LoRA ($r = 16$, $\alpha = 32$) across its language, vision, and merger modules, plus the <R> embedding and the rotation head (55.1M parameters, 0.62%). We set $K = 128$, $\delta = 10°$, and $\lambda_{\text{coarse}} = \lambda_{\text{res}} = 0.2$, and train 10 epochs with AdamW on eight A100 GPUs (global batch 32, BF16, cosine decay $10^{-4} \to 10^{-5}$). Inference is greedy by default; remaining details are in Appendix B.

Datasets and Task Protocols. Training uses 60K images from FoundationPose (Wen et al., 2024) and 30K each from Objectron (Ahmadyan et al., 2021) and HOT3D (Banerjee et al., 2025), while One2Any (Liu et al., 2025a) and FoundationPose train on the 1.2M images. Without target-dataset fine-tuning, one checkpoint covers the official test sets: single-reference LINEMOD, LM-O (Hinterstoisser et al., 2011; Brachmann et al., 2014), YCB-Video (Xiang et al., 2017), TUD-L, IC-BIN, HB (Hodan et al., 2018), and T-LESS (Hodan et al., 2017); 16-reference LINEMOD and YCB-Video; and YCBInEOAT (Wen et al., 2021) and YCB-Video tracking. Evaluation uses real references without CAD models; all protocol differences are marked in tables.

Evaluation Metrics. ADD is the mean corresponding-point distance between predicted and ground-truth poses; ADD-S uses closest points for symmetric objects. ADD-0.1d reports the percentage within 10% of the object diameter; ADD/ADD-S AUC integrates accuracy over distance thresholds; BOP Average Recall (AR) averages symmetry-aware VSD, MSSD, and MSPD recall. We report ADD-0.1d on LINEMOD, ADD/ADD-S AUC on YCB-Video and tracking, and BOP AR on the other benchmarks and in the aggregate YCB-Video comparison.

4.2 Comparisons with Baselines

Qualitative Results. Figure 5 compares one PoseVLM checkpoint with OneViewAll (Luo et al., 2026) for single-reference estimation, FoundationPose (Wen et al., 2024) for multi-reference estimation, and BundleTrack (Wen & Bekris, 2021) for tracking. PoseVLM remains aligned under occlusion and symmetry, resolves viewpoint variation from 16 references, and recovers from tracking drift, matching strong baselines across all three settings.

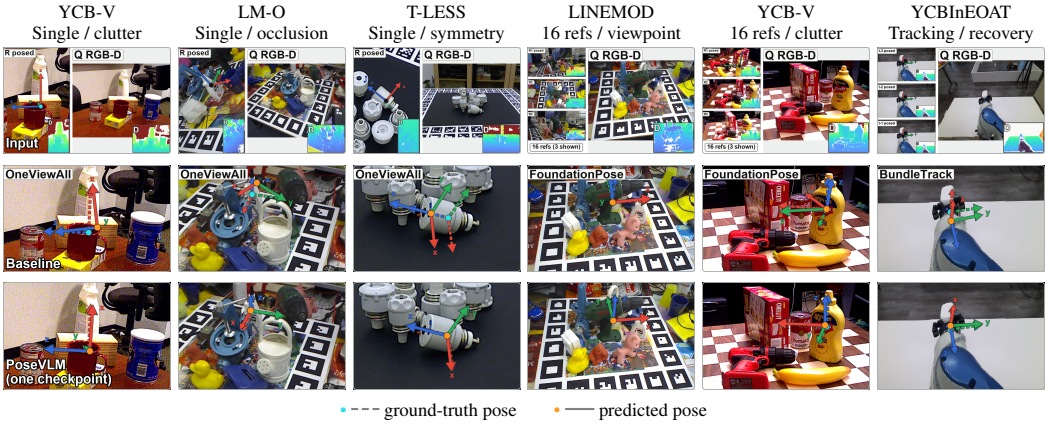

Figure 5: Qualitative comparison with representative baselines OneViewAll (Luo et al., 2026), FoundationPose (Wen et al., 2024), and BundleTrack (Wen & Bekris, 2021). Dashed triads with cyan origins denote ground-truth poses and solid triads with orange origins denote predictions; for PoseVLM the two nearly coincide.

Across one or 16 real references and recent predictions, one checkpoint requires no object-specific models, target-dataset fine-tuning, or test-time refinement; Appendix K provides object-level results.

Single-Reference Pose Estimation. This setting tests novel-object generalization. Tables 1 and 2 show that PoseVLM outperforms single-reference methods on all six evaluated BOP datasets, and achieves the strongest aggregate LINEMOD and YCB-Video results without CAD models or ground-truth pose components.

Table 1: Cross-dataset BOP AR (%) of LM-O, TUD-L, IC-BIN, HB, YCB-V, and T-LESS. PoseVLM* receives only RGB-D renders from the CAD used by FoundationPose.

Method	Object prior	Reference	LM-O	TUD-L	IC-BIN	HB	YCB-V	T-LESS
Additional CAD object prior								
MegaPose (Labbé et al., 2022)	CAD	rendered set	58.3	71.2	37.1	75.7	63.3	54.3
SAM-6D (Lin et al., 2024)	CAD	rendered set	12.9	37.9	11.2	25.2	22.4	–
ZeroPose (Chen et al., 2023)	CAD	rendered set	56.2	87.2	41.8	68.2	58.4	–
FoundationPose (Wen et al., 2024)	CAD	rendered set	78.8	–	–	–	88.0	83.0
PoseVLM*	CAD	1 rendered	**81.5**	**93.2**	**62.8**	**85.7**	**93.8**	**89.4**
Single reference without CAD								
SinRef-6D (Liu et al., 2026)	none	1 rendered	61.8	88.9	44.0	63.3	65.1	67.0
OneViewAll (Luo et al., 2026)	none	1 rendered	66.4	90.1	51.2	76.8	74.0	82.4
PoseVLM	none	1 real	**68.1**	**91.7**	**56.4**	**79.2**	**86.4**	**87.1**

Table 2: Single-reference comparison (%): ADD-0.1d on LINEMOD and ADD/ADD-S on YCB-V. Extra denotes test-time information; OneViewAll reports $N = 78$ pose hypotheses per query.

Method	Input	Extra	Reference	LM	YCB-V ADD	YCB-V ADD-S
Additional object asset or pose information						
FoundationPose (Wen et al., 2024)	RGB-D	CAD	1 real	48.3	76.1	90.4
NOPE (Nguyen et al., 2024)	RGB	GT-t	1 real	16.3	25.1	86.0
Single reference without CAD or GT pose components						
Oryon (Corsetti et al., 2024)	RGB-D	none	1 real	9.8	7.4	13.3
One2Any (Liu et al., 2025a)	RGB-D	none	1 real	52.6	84.4	93.7
CoordAR (Zuo et al., 2026)	RGB-D	none	1 real	75.0	78.5	95.5
SinRef-6D (Liu et al., 2026)	RGB-D	none	1 rendered	90.3	52.8	–
OneViewAll (N=78) (Luo et al., 2026)	RGB-D	none	1 real	92.5	84.4	89.9
PoseVLM	RGB-D	none	1 real	**94.7**	**87.6**	**95.8**

Multi-Reference Pose Estimation. Table 3 evaluates the same checkpoint with 16 reference views. PoseVLM matches the near-saturated LINEMOD performance of FoundationPose and establishes the strongest YCB-Video results, while consuming the references directly rather than reconstructing an instance-specific object model. All three metrics improve over single-reference inference, showing that the shared interface integrates complementary observations without a multi-view pipeline.

Table 3: Aggregate 16-reference comparison (%): ADD-0.1d on LINEMOD and ADD/ADD-S AUC on YCB-V. FT: target-dataset fine-tuning; recon.: reconstruction from reference views.

Method	Input	Object asset	Ref.	Target FT	LM	YCB-V ADD	YCB-V ADD-S
PREDATOR (Huang et al., 2021)	RGB-D	none	16	Yes (YCB-V)	–	24.3	71.0
LoFTR (Sun et al., 2021)	RGB-D	none	16	Yes (YCB-V)	–	26.2	52.5
LatentFusion (Park et al., 2020)	RGB-D	none	16	No	87.1	–	–
FS6D (He et al., 2022)	RGB-D	none	16	Yes	88.9	42.1	88.4
FoundationPose (Wen et al., 2024)	RGB-D	instance recon.	16	No	**99.9**	91.5	97.4
PoseVLM	RGB-D	none	16	No	99.6	**92.3**	**98.5**

Pose Tracking. Tracking tests whether PoseVLM can turn its own predictions into reliable reference evidence: only the first frame receives ground-truth pose; PoseVLM reuses three latest predictions without re-initialization. Under every matched evidence setting in Table 4, PoseVLM establishes the best ADD and ADD-S on both datasets; with real-reference or predicted-history evidence alone, PoseVLM requires neither CAD model, nor online reconstruction, nor a tracking-specific pipeline. Performance rises from a fixed initial reference to one and then three recent predictions, showing that refreshed short-term history strengthens recursive inference.

Table 4: Tracking and history ablation using ADD/ADD-S AUC (%), grouped by test-time object evidence. All rows use GT initialization without re-initialization; bold marks each setting's best, and PoseVLM* uses matched FoundationPose RGB-D renders.

				YCBInEOAT		YCB-Video	
Method / variant	Object evidence	Init.	Reinit.	ADD	ADD-S	ADD	ADD-S
Pre-existing CAD object model							
se(3)-TrackNet (Wen et al., 2021)	CAD	GT	No	92.66	95.53	93.0	95.7
Wüthrich (Wüthrich et al., 2013)	CAD	GT	No	78.28	89.18	78.0	90.2
RGF (Issac et al., 2016)	CAD	GT	No	29.98	39.90	59.2	74.3
ICG (Stoiber et al., 2022)	CAD	GT	No	–	–	86.4	96.5
FoundationPose (model) (Wen et al., 2024)	CAD	GT	No	93.09	96.42	96.0	97.9
PoseVLM*	CAD renders	GT	No	**93.86**	**98.91**	**96.6**	**98.7**
No pre-existing CAD model							
BundleTrack (Wen & Bekris, 2021)	online recon.	GT	No	87.34	92.53	–	–
BundleSDF (Wen et al., 2023)	online recon.	GT	No	86.95	93.77	–	–
FoundationPose (16 ref.) (Wen et al., 2024)	instance recon.	GT	No	–	–	93.7	97.5
FoundationPose (1st ref.) (Wen et al., 2024)	instance recon.	GT	No	–	–	57.9	80.9
One2Any (Liu et al., 2025a)	1st real	GT	No	–	–	84.8	93.8
CoordAR (Zuo et al., 2026)	1st real + GT mask	GT	No	–	–	87.1	95.8
PoseVLM* (16 ref.)	recon. renders	GT	No	**93.35**	**98.64**	**93.9**	**98.8**
PoseVLM history evidence (same checkpoint)							
PoseVLM (initial only)	1 fixed real	GT	No	89.62	96.54	87.6	95.8
PoseVLM (1 recent)	1 prediction	GT	No	91.25	97.18	89.3	96.6
PoseVLM (3 recent)	3 predictions	GT	No	**93.17**	**98.68**	**92.3**	**98.5**

4.3 Ablation Studies

We evaluate PoseVLM's core components and inference variants under a common LINEMOD protocol. Beyond aggregate ADD-0.1d, the component-wise errors in Table 5 reveal a clear division of labor, Figure 6 examines backbone, test-time scaling and latency, and the **Appendix** further covers why pose-specific adaption is needed and other experiments.

Multiple Coarse Rotation Targets. We replace the multi-target coarse supervision with a single target while keeping the data and optimization budget fixed. Rotation deteriorates sharply while translation remains essentially unchanged, showing that a single label cannot consistently represent neighboring and symmetry-equivalent bins. Retaining all equally correct targets therefore resolves a rotation-specific supervision ambiguity; Appendix G separates the two target sources (Table 9).

Coarse-to-Fine Rotation Decoding. Removing the coarse-to-fine factorization produces the same component-wise pattern: translation is preserved, but rotation degrades substantially. This isolates the benefit to rotation decoding and supports decomposing it into global orientation selection and local residual refinement.

Table 5: **Ablations.** Rot. err. and Trans. err. are the mean rotation error (°) and mean translation error (mm).

Variant	LM ADD-0.1d	Rot. err. (°) ↓	Trans. err. (mm) ↓	Latency (s)
Full model	94.7	4.95	7.61	0.07
Rotation decoding				
w/o multiple coarse targets	57.5	19.35	7.49	0.07
w/o coarse-to-fine rotation	52.7	21.51	8.43	0.07
Translation decoding				
w/o z-CoT	75.9	4.91	13.35	0.05
w/o geometry grounding	87.3	4.97	10.14	0.06
Dual-branch outputs				
w/o rotation head	43.6	23.42	7.57	0.07
w/o text decoding	62.8	4.69	17.48	0.05

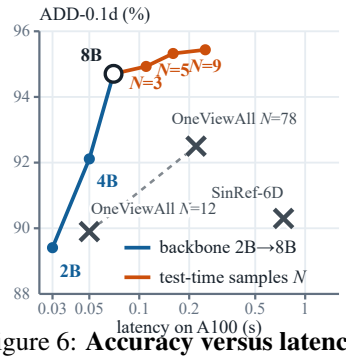

Figure 6: **Accuracy versus latency on LINEMOD.** Backbone and test-time scaling of PoseVLM; × marks single-reference baselines.

Geometry-Grounded z-CoT. We compare direct decoding, VLM-generated free-form reasoning text without explicit metric geometry, and z-CoT. Their rotation errors remain nearly identical, while translation improves progressively across the three variants. Generic rationales recover only part of this improvement, indicating that the key benefit comes from explicit metric grounding rather than additional reasoning tokens alone. Table 6 decomposes the translation error into its depth and in-plane components, which shows t_z is the major error source and z-CoT improves t_z substantially. Appendix I further measures the per-step accuracy and their correlation with the final pose.

Table 6: **Translation error decomposition.** t_z err. and t_x, t_y err. are the mean depth and in-plane translation errors (mm).

	w/o z-CoT	w/o geometry grounding	**Full model**
t_z err. (mm) ↓	11.72	8.04	**4.62**
t_x, t_y err. (mm) ↓	5.59	5.53	**5.47**

Dual-Branch Outputs. We compare text-only decoding, head-only prediction, and their combination to test the intended division of labor directly. The component-wise results show that the text pathway provides the stronger translation estimate, while the dedicated head corrects its weaker rotation prediction; their combination retains both strengths.

Backbone Scaling. Figure 6 scales Qwen3-VL across 2B, 4B, and 8B while fixing PoseVLM's design, data, training, input evidence, supervision, and heads. The monotonic ADD gains therefore reflect backbone capacity alone, at the cost of higher latency; other experiments use 8B.

Test-Time Scaling. For optional test-time scaling (Figure 6), we keep the rotation fixed and aggregate the coordinate-wise median of N back-projected translations from one greedy generation and $N-1$ sampled language generations. More samples improve pose accuracy, providing an accuracy–latency trade-off; all main results use greedy inference.

Efficiency. Figure 6 also compares latency with single-reference baselines on the same A100. PoseVLM's greedy decoding (94.7% at 0.07 s) is faster and more accurate than OneViewAll's full configuration (92.5% at 0.22 s) and than SinRef-6D (90.3% at 0.74 s); the only faster pipelines, OneViewAll (N=12, 0.05 s) and One2Any (0.06 s), trail by 4.8 and 42.1 points. PoseVLM decodes a pose directly, whereas these pipelines score and iteratively refine many hypotheses.

5 Conclusion

We presented PoseVLM, a unified VLM that directly estimates metric 6DoF object pose across single-reference estimation, multi-reference estimation, and tracking. PoseVLM assigns each pose component to the decoding path that suits it. Its rotation branch selects a coarse orientation, refines it with a bin-conditioned residual, and is supervised to accept nearby and symmetry-equivalent coarse targets. Its language branch recovers object-center depth from visible surface depth and cross-view geometry through geometry-grounded z-CoT. With a single checkpoint, PoseVLM leads model-free single-reference methods, achieves the strongest 16-reference results on YCB-Video, and establishes the best reported tracking ADD and ADD-S on both evaluated datasets under matched object-evidence settings, without per-instance reconstruction or task-specific pipelines. These results show that VLMs can move beyond auxiliary cue generation and directly recover accurate metric pose by matching supervision and decoding to the geometry of each pose component.

AI USE STATEMENT

In this work, we used generative AI tools for drafting and polishing the manuscript text. We have not used generative AI tools for writing experimental code, running or modifying experiments, or producing figures and tables, and the remaining disclosure categories are not applicable to this work. All AI-assisted text was reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7822–7831, 2021.

Kahina Amara, Mohamed Amine Guerroudji, Nadia Zenati, Oussama Kerdjidj, Shadi Atalla, Wathiq Mansoor, and Naeem Ramzan. Augmented reality localisation using 6 dof phantom head pose estimation-based generalisable deep learning model. In *2024 8th International conference on image and signal processing and their applications (ISPA)*, pp. 1–6. IEEE, 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, et al. Hot3d: Hand and object tracking in 3d from egocentric multi-view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7061–7071, 2025.

Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pp. 536–551. Springer, 2014.

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.

Zhipeng Cai, Ching-Feng Yeh, Hu Xu, Zhuang Liu, Gregory Meyer, Xinjie Lei, Changsheng Zhao, Shang-Wen Li, Vikas Chandra, and Yangyang Shi. Depthlm: Metric depth from vision language models. *arXiv preprint arXiv:2509.25413*, 2025.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024a.

Jianqiu Chen, Zikun Zhou, Mingshan Sun, Tianpeng Bao, Rui Zhao, Liwei Wu, and Zhenyu He. Zeropose: Cad-prompted zero-shot object 6d pose estimation in cluttered scenes. *arXiv preprint arXiv:2305.17934*, 2023.

Kai Chen, Yiyao Ma, Xingyu Lin, Stephen James, Jianshu Zhou, Yun-Hui Liu, Pieter Abbeel, and Qi Dou. Vision foundation model enables generalizable object pose estimation. *Advances in Neural Information Processing Systems*, 37:19975–20002, 2024b.

Wei Chen, Xi Jia, Hyung Jin Chang, Jinming Duan, Linlin Shen, and Ales Leonardis. Fs-net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1581–1590, 2021.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024.

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.

Jaime Corsetti, Davide Boscaini, Francesco Giuliari, Changjae Oh, Andrea Cavallaro, and Fabio Poiesi. High-resolution open-vocabulary object 6d pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking. *IEEE Transactions on Robotics*, 37 (5):1328–1342, 2021.

Yan Di, Ruida Zhang, Zhiqiang Lou, Fabian Manhardt, Xiangyang Ji, Nassir Navab, and Federico Tombari. Gpv-pose: Category-level object pose estimation via geometry-guided point-wise voting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6781–6791, 2022.

Yufei Ding, Haoran Geng, Chaoyi Xu, Xiaomeng Fang, Jiazhao Zhang, Songlin Wei, Qiyu Dai, Zhizheng Zhang, and He Wang. Open6dor: Benchmarking open-instruction 6-dof object rearrangement and a vlm-based approach. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7359–7366. IEEE, 2024.

Zhiwen Fan, Panwang Pan, Peihao Wang, Yifan Jiang, Dejia Xu, and Zhangyang Wang. Pope: 6-dof promptable pose estimation of any object in any scene with one reference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7771–7781, 2024.

Xueyang Guo, Hongwei Hu, Chengye Song, Jiale Chen, Zilin Zhao, Yu Fu, Bowen Guan, and Zhenze Liu. Unidiffgrasp: A unified framework integrating vlm reasoning and vlm-guided part diffusion for open-vocabulary constrained grasping with dual arms. *arXiv preprint arXiv:2505.06832*, 2025.

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, 2022.

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 Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pp. 880–888. IEEE, 2017.

Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 19–34, 2018.

Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 4267–4276, 2021.

Jan Issac, Manuel Wüthrich, Cristina Garcia Cifuentes, Jeannette Bohg, Sebastian Trimpe, and Stefan Schaal. Depth-based object tracking using a robust gaussian filter. In *2016 IEEE international conference on robotics and automation (ICRA)*, pp. 608–615. IEEE, 2016.

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

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. *arXiv preprint arXiv:2212.06870*, 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.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

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*, pp. 27906–27916, 2024.

Jian Liu, Wei Sun, Kai Zeng, Jin Zheng, Hui Yang, Hossein Rahmani, Ajmal Mian, and Lin Wang. Scalable unseen objects 6-dof absolute pose estimation with robotic integration. *IEEE Transactions on Robotics*, 2026.

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.

Sheng Liu, Zhe Li, Weiheng Wang, Han Sun, Heng Zhang, Hongpeng Chen, Yusen Qin, Arash Ajoudani, and Yizhao Wang. Activepose: Active 6d object pose estimation and tracking for robotic manipulation. *arXiv preprint arXiv:2509.11364*, 2025b.

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 *European Conference on Computer Vision*, pp. 298–315. Springer, 2022.

Zibin Liu, Banglei Guan, Yang Shang, Shunkun Liang, Zhenbao Yu, and Qifeng Yu. Optical flow-guided 6dof object pose tracking with an event camera. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 6501–6509, 2024a.

Zibin Liu, Banglei Guan, Yang Shang, Qifeng Yu, and Laurent Kneip. Line-based 6-dof object pose estimation and tracking with an event camera. *IEEE Transactions on Image Processing*, 33: 4765–4780, 2024b.

Yang Luo, Yan Gong, Yongsheng Gao, Jie Zhao, Xinyu Zhang, and Huaping Liu. Oneviewall: Semantic prior guided one-view 6d pose estimation for novel objects. *arXiv preprint arXiv:2605.07023*, 2026.

Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14420–14431, 2024.

Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Yinlin Hu, Renaud Marlet, Mathieu Salzmann, and Vincent Lepetit. Nope: Novel object pose estimation from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17923–17932, 2024.

OpenAI. GPT-5.6 Sol Model. `https://developers.openai.com/api/docs/models/gpt-5.6-sol`, 2026.

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*, pp. 163–182. Springer, 2024.

Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10710–10719, 2020.

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.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026. URL https://qwen.ai/blog?id=qwen3.6-27b.

Manuel Stoiber, Martin Sundermeyer, and Rudolph Triebel. Iterative corresponding geometry: Fusing region and depth for highly efficient 3d tracking of textureless objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6855–6865, 2022.

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, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8922–8931, 2021.

Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. Onepose: One-shot object pose estimation without cad models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6825–6834, 2022.

Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16611–16621, 2021.

He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2642–2651, 2019.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Bowen Wen and Kostas Bekris. Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8067–8074. IEEE, 2021.

Bowen Wen, Chaitanya Mitash, and Kostas Bekris. Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. *arXiv preprint arXiv:2105.14391*, 2021.

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

Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 606–617, 2023.

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.

Manuel Wüthrich, Peter Pastor, Mrinal Kalakrishnan, Jeannette Bohg, and Stefan Schaal. Probabilistic object tracking using a range camera. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pp. 3195–3202. IEEE, 2013.

Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

Tomoya Yoshida, Shuhei Kurita, Taichi Nishimura, and Shinsuke Mori. Generating 6dof object manipulation trajectories from action description in egocentric vision. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17370–17382, 2025.

Hanxun Yu, Xuan Qu, Yuxin Wang, Jianke Zhu, and Lei Ke. Unlocking dense metric depth estimation in vlms. *arXiv preprint arXiv:2605.15876*, 2026.

Jiyao Zhang, Mingdong Wu, and Hao Dong. Genpose: Generative category-level object pose estimation via diffusion models. *arXiv preprint arXiv:2306.10531*, 2023a.

Ruihang Zhang, Felix Taubner, Pooja Ravi, Kiriakos N Kutulakos, and David B Lindell. Proxypose: 6-dof pose tracking via video-to-video translation. *arXiv preprint arXiv:2607.06555*, 2026.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023b.

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 Limitations and Future Work

Input requirements and depth reliability. PoseVLM requires RGB-D observations, known camera intrinsics, and a posed first reference to define the canonical object frame. It does not estimate this initial reference pose from scratch. Its z-CoT formulation relies on visible surface depth and a signed surface-to-center offset, so missing or biased depth may compromise the evidence used for metric translation. Although predicted depth is included in the training data, the current evaluation does not systematically isolate sensitivity to depth corruption or reference-pose errors. Future work could incorporate depth-confidence cues and uncertainty-aware geometric reasoning to better handle imperfect observations.

Reference selection. The current protocols use predetermined single references or uniformly sampled multi-reference views. An important direction is adaptive reference selection that considers visibility, viewpoint complementarity, and pose confidence, aiming to retain useful geometric evidence without unnecessarily expanding the visual context.

Long-term tracking and recovery. PoseVLM performs recursive inference using recent predicted poses, with ground-truth initialization only in the first frame. Training with perturbed history poses exposes the model to imperfect references, and the history ablation supports the benefit of refreshed observations. However, the current results do not constitute a dedicated evaluation of recovery after prolonged complete occlusion or extended tracking loss. If several consecutive predictions become unreliable, using them as subsequent references may propagate errors. Future work could combine short-term history with persistent reliable keyframes, uncertainty-aware history updates, and explicit relocalization to improve recovery while retaining the shared pose predictor.

B Implementation and Training Details

This section complements the summary in Section 4.1 with the training composition, supervision details, and inference protocol.

Optimization. PoseVLM freezes Qwen3-VL-8B-Instruct (Bai et al., 2025) and trains LoRA adapters (r=16, α=32) on the language, vision, and merger modules, together with the newly added `<R>` embedding and the rotation head (55.1M parameters, 0.62% of all parameters). We train 10 epochs with AdamW on eight A100 GPUs (global batch 32, BF16, cosine decay $10^{-4} \rightarrow 10^{-5}$) with loss weights $\lambda_{\text{coarse}} = \lambda_{\text{res}} = 0.2$.

Training Data Composition. Training uses 60K FoundationPose (Wen et al., 2024) images, 30K Objectron (Ahmadyan et al., 2021) images, and 30K HOT3D (Banerjee et al., 2025) images. Each example belongs to one of the three settings: single-reference examples contain one posed reference observation; multi-reference examples add a random number of unposed reference observations; tracking examples use a random number of previous frames with noise-perturbed poses to simulate accumulated drift. The three settings are roughly balanced in the training mixture. Queries are always full frames without object cues; each reference observation appears as a full frame or as an object crop with equal probability.

Supervision Data.

The three training data sources differ in how the supervision of Appendix H is constructed. FoundationPose examples are synthetic renders, so their depth, object meshes, masks, and visibility are exact by construction. HOT3D provides official high-resolution object scans together with pose annotations, while Objectron provides official pose annotations (3D boxes) but no object models; we therefore reconstruct one Objectron mesh per object offline from the video and its sparse point cloud. On both datasets, object masks and visibility are obtained by projecting the object meshes with the annotated poses. Neither source provides dense depth, so we predict metric depth for both query and reference frames with Depth Anything 3 (Lin et al., 2025).

Rotation Bins and Symmetry Discretization. The $K{=}128$ rotation-bin centers are selected by farthest-point sampling from 20,000 uniformly random rotations under the $\mathrm{SO}(3)$ geodesic distance, which spreads them evenly over the rotation space, and are shared across all objects, datasets, and tasks. Object symmetries follow the BOP definitions; continuous rotational symmetries are discretized at a fixed angular interval of 45°.

Inference. All main results use greedy decoding, and the rotation branch selects the maximum-probability bin. Latency is measured end-to-end per query on a single A100 with batch size 1, including prompt encoding and the full z-CoT generation; the N test-time samples are decoded sequentially and share the prompt key-value cache.

C Evaluation Protocols

This section details the protocol for each reference regime. Baseline results are quoted from the original papers when available and otherwise reproduced under the same protocol.

Single-Reference Protocol. For each object, PoseVLM receives one real RGB-D reference frame with ground-truth pose. On the benchmarks evaluated by OneViewAll and SinRef-6D, we use the same reference selection as these methods; on the remaining benchmarks, we sample one reference frame per object with a fixed random seed. PoseVLM* denotes the same checkpoint prompted with rendered RGB-D references instead: one render sampled from the rendered set used by FoundationPose in Table 1 and Table 4 (top), and renders of the 16-view instance reconstruction in Table 4 (middle).

Multi-Reference Protocol. The 16 reference views are sampled uniformly from the posed reference sequence; only the first reference is encoded with its pose, and the remaining views are provided as RGB-D only.

Tracking Protocol. Only the first frame receives ground-truth pose. At each frame, the prompt contains the K latest predictions with $K \in \{0, 1, 3\}$; until K predictions exist, the initial posed reference fills the remaining slots. The same model runs frame by frame and uses only this evidence.

D Prompt and Output Format

A single-reference prompt contains four images and a text block: (A) the reference RGB image with the coordinate triad (+X red, +Y green, +Z blue), (B) the reference depth rendered as color, (C) the full query RGB frame, and (D) the query depth rendered as color, all with the fixed color-to-depth scale. The text provides the reference center depth and camera intrinsics, and it fixes the z-CoT reasoning path for the language branch:

> Images: A=reference RGB with the coordinate triad (+X red, +Y green, +Z blue); B=reference depth as color; C=query RGB; D=query depth as color; all depth maps are normalized to [0,1] via mapping 421 mm to 0 and 1389 mm to 1.. Ref uvz(mm): O=(559,441,894); intrinsics fx=1000, fy=1000, cx=W/2, cy=H/2. Origin coordinates use the [0,1000] normalized convention of Image C. CoT task: first reason about the query object-center depth, then return the final origin. Use one fixed reasoning path for z: center depth = visible surface depth + signed surface-to-center offset. Do not rely on the single depth value at the center pixel, because the projected center may lie inside the object or be occluded. Instead, inspect D over the visible object region and estimate a stable visible surface depth from the closest depth band. Then use the axes in Image A, the object shape, and the 3D reference-to-query view change to estimate the signed surface-to-center offset from that surface point to the object center. Add the offset to the surface depth to get the final center depth z_mm. Return exactly this block: <origin>u,v,z_mm</origin>Rules: u,v integers in [0,1000]; z_mm camera depth in millimeters; no spaces inside the origin tag.

The language branch generates the four z-CoT steps and ends with the translation span:

> The object center in the first reference is at $(559, 441)$ with a depth of 894 mm. In the query, the projected center is at $(134, 276)$, and a visible surface point selected near that location has an observed depth of 804 mm. The same surface point is grounded at $(568, 427)$ in the reference, where its geometry relative to the canonical center implies a signed surface-to-center offset of -19 mm along the query depth axis. Therefore, the query center depth is $t_z = 804 + (-19) = 785$ mm. <origin>134,276,785</origin>

The rotation head decodes the rotation in parallel from the <R> hidden state in one forward pass.

E Diagnostic Experiments

Before full-scale training, we run a series of small-scale experiments on LINEMOD to identify which pose components a VLM can learn. All variants are trained on the same 10K-image LINEMOD subset with the same schedule as the full model, so their results reflect how easily each output can be learned. These early experiments and the hyperparameter sweeps in Appendix F are evaluated on a validation split of 10% of the LINEMOD training images, held out from training and disjoint from the official test set.

Language-Decoder Capability Tests. As shown in Figure 7, directly fine-tuning the VLM to generate the full pose as text, token by token, yields only 2.6% ADD-$0.1d$. We next isolate the two outputs by training the VLM to generate only one pose component and fixing the other to its ground truth during evaluation. The rotation-only variant reaches only 3.0% ADD-$0.1d$, whereas the translation-only variant attains 80.4%.

A complementary test replaces the language decoder with a dedicated head that predicts the same 2D center projection and depth and back-projects them; it reaches only 54.6%, so translation stays in the language branch.

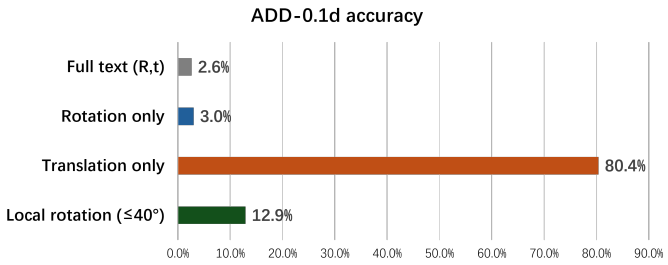

Figure 7: **Language-Decoder Capability Tests.** Text generation handles translation much better than rotation, while discrete selection and local rotation prediction remain learnable. Each test uses ground truth for the component that the variant does not predict.

We then ask the VLM to identify the correct rotation among 32 candidates. Its 92.3% accuracy shows that the VLM can reliably pick the correct global orientation from candidates, even though it fails to generate continuous rotations directly. Finally, we constrain the angular difference between the reference and query poses to at most 40° and ask the VLM to generate only rotation. This local setting reaches 12.9% ADD-$0.1d$, more than four times the 3.0% obtained by unrestricted rotation generation. Together, these tests show two complementary abilities: picking the correct global orientation from candidates, and refining a rotation that is already close.

Rotation-Decoder Diagnosis. Coarse-to-fine rotation allows several bin–residual pairs to be equally correct for the same target. Under single-target text supervision, the model selects the best possible bin in only 22.4% of cases. If we also accept any prediction whose rotation error is within 10° of the best possible bin's error, the accuracy rises to 54.2%. This gap shows that many of the remaining classification errors point to bins that are geometrically close to the best one. Giving the model multiple text targets for the same input degrades both measures further: exact best-bin accuracy drops to 13.2%, and accuracy under the 10° tolerance falls to 32.9%. Consequently, the single-target and multi-target text variants attain only 16.1% and 10.2% rotation-only ADD-$0.1d$.

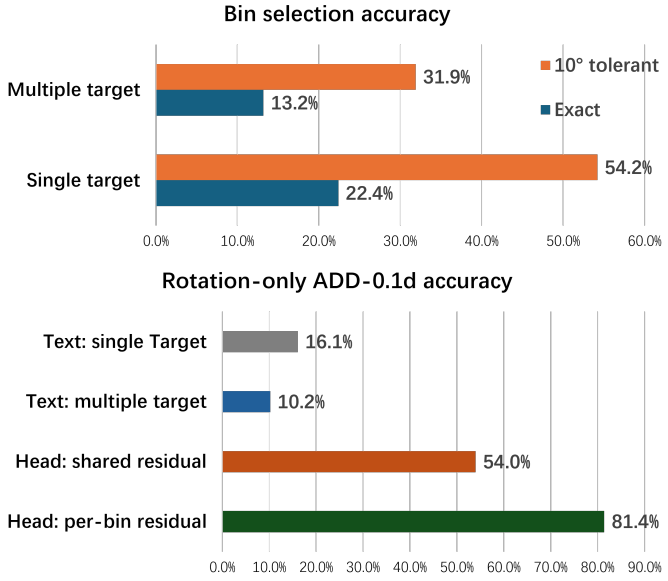

Figure 8: **Rotation-Decoder Diagnosis.** Text-based coarse-to-fine targets remain ambiguous. A probability head improves rotation prediction, and giving every bin its own residual performs best.

Moving the multi-target coarse prediction to a dedicated head raises rotation-only ADD-$0.1d$ to 54.0% when a single residual is shared across bins. Each bin needs a different correction, so the shared residual cannot fit them all. Giving every bin its own residual raises rotation-only ADD-$0.1d$ further to 81.4%.

Ground-Truth Substitution Diagnosis. After improving the rotation decoder, we locate the bottleneck of pose accuracy by replacing one predicted component with its ground truth at a time. On the checkpoint used for this analysis, predicting the full pose gives 70.8% ADD-$0.1d$. Replacing translation with its ground truth raises accuracy to 91.1%, whereas replacing rotation reaches 81.2%, corresponding to gains of 20.3 and 10.4 percentage points. Within translation, replacing only t_z with ground truth raises accuracy to 89.2%, while replacing t_x and t_y together reaches only 74.7%. The dominant remaining error is therefore concentrated along the depth axis. Section 3.3 of the main paper reports the same substitution protocol on the final model's direct-decoding variant (75.9% ADD-$0.1d$ baseline); the numbers here use an earlier diagnostic checkpoint, so the baselines differ.

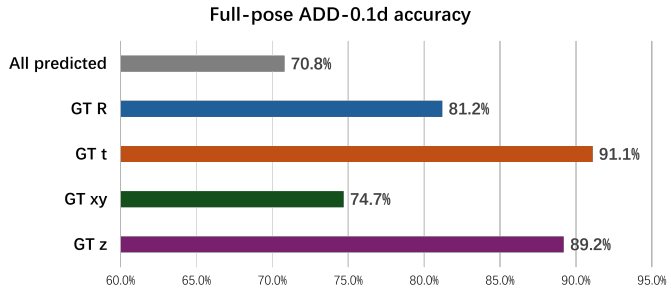

Figure 9: **Ground-Truth Substitution Diagnosis.** Replacing predicted components with ground truth identifies translation, and specifically its depth component t_z, as the dominant error source.

Comparison with Off-the-Shelf VLMs. Table 7 evaluates generic instruction-following VLMs under the same single-reference protocol as the main paper: each model receives the same reference-and-query prompt as PoseVLM, and we parse the first coordinate span from its output and back-project it. We report ADD AUC because ADD-$0.1d$ is close to zero for zero-shot models. Explicit thinking modes and larger generic VLMs remain far behind, confirming that metric pose estimation needs the adapted visual-language representation and decoding objectives of PoseVLM.

Table 7: Pose estimation with off-the-shelf VLMs (LINEMOD ADD AUC).

Model	LM ADD AUC
Qwen3-VL-8B-Instruct (Bai et al., 2025)	4.16
Qwen3-VL-8B-Thinking (Bai et al., 2025)	0.04
Qwen3.6-27B (Qwen Team, 2026)	0.00
GPT-5.6 Sol (OpenAI, 2026)	4.78
PoseVLM (8B)	**92.4**

F Hyperparameter Selection

We selected the number of rotation bins $K = 128$ and the angular tolerance $\delta = 10°$ in early experiments before full-scale training, then kept them fixed throughout the subsequent full-scale experiments; the sweeps use the validation split described in Appendix E. The two parameters control complementary aspects of rotation decoding: K sets the granularity of the coarse orientations, while δ determines how far a bin may lie beyond the best match and still receive supervision. We evaluated the two parameters separately, testing $K \in \{32, 128, 512, 1024\}$ and $\delta \in \{0°, 5°, 10°, 20°\}$, as illustrated in Table 8.

Table 8: **Early selection of rotation hyperparameters.** The number of rotation bins K (left) and the angular tolerance δ (right) were evaluated in early sweeps, each with the other parameter fixed. Selected values are marked in the table and fixed for subsequent full-scale training.

K	Rot. err. (°) ↓	ADD-$0.1d$ (%) ↑
32	19.88	57.1
128 (selected)	5.02	83.6
512	5.08	83.3
1024	5.22	81.0

δ	Rot. err. (°) ↓	ADD-$0.1d$ (%) ↑
0°	17.94	58.9
5°	10.36	74.2
10° (selected)	5.02	83.6
20°	5.13	82.4

G Ablation Details

All variants in Table 5 and Table 9 are retrained from scratch with the same data and schedule as the full model and are evaluated under the same single-reference LINEMOD protocol.

Rotation decoding. **w/o multiple coarse targets** trains the coarse selection with only the bin nearest to the target rotation, dropping the equally correct neighboring and symmetry-related bins. **w/o coarse-to-fine rotation** removes the bin–residual factorization and predicts rotation by direct 6D regression from the <R> hidden state.

Multi-target decomposition. Because **w/o multiple coarse targets** in Table 5 removes the two target types together, we further test variants remove one source at a time in the Table 9. **w/o neighbor tolerance** sets δ=0: for each symmetry-equivalent target $\mathbf{R}^*\mathbf{S}$, only the bin nearest to that target remains in $\mathcal{C}(\mathbf{R}^*)$ (i.e., the per-target argmin over bins, taken separately for each $\mathbf{S}$ and unioned); for asymmetric objects this leaves a single nearest bin, and for symmetric objects one nearest bin per symmetry-equivalent rotation. This also corresponds to the δ=0° setting of the sweep in Table 8. **w/o symmetry-equivalent targets** keeps the δ-neighborhood but supervises only bins near the target rotation itself, so symmetric objects receive a single canonical target. The two factors address distinct failure modes: the δ-neighborhood absorbs bin discretization, since the target rotation generally lies between bin centers, while the symmetry-equivalent targets absorb physical symmetry, whose orientations are visually indistinguishable. Table 9 shows that both factors are necessary for the best performance.

Table 9: **Multi-target decomposition.** ADD-0.1d (%) on LINEMOD when each source of equally correct coarse targets is removed. "Symmetric objects" is the mean over eggbox and glue; "asymmetric objects" is the mean over the remaining 11 objects.

Variant	Overall	Asymmetric obj.	Symmetric obj.
Full model	94.7	93.8	100.0
w/o neighbor tolerance (δ=0)	61.8	60.4	69.6
w/o symmetry-equivalent targets	93.8	93.1	97.5
Single target (both removed)	57.5	55.7	67.5

Translation decoding. **w/o z-CoT** decodes the translation in a single step, skipping the reasoning chain. **w/o geometry grounding** replaces the z-CoT targets with free-form rationales generated zero-shot by a pretrained VLM, which serve as the training targets of the language branch. This control changes both the supervision source and the text structure, so it evaluates the complete z-CoT training scheme.

Dual-branch outputs. **w/o rotation head** decodes rotation as text, generating the bin index followed by the residual. **w/o text decoding** predicts translation with the same 2D-projection-and-depth head as the capability tests in Appendix E.

H z-CoT Target Construction

The z-CoT supervision is constructed offline for each training query from the ground-truth query pose and the object's CAD mesh.

Surface Depth and Reference Correspondence. Let (u, v) be the projection of the canonical object center O in the query image. Among the depth samples near (u, v) that fall on the target object according to its ground-truth segmentation mask and remain visible in a rendering of the object mesh at the ground-truth query pose, we take the one nearest to (u, v), at pixel (u_s, v_s) with depth z_{surf}; queries with no valid sample are discarded. Back-projecting z_{surf} along the ray through its own pixel (u_s, v_s) yields the visible surface point P, which lies on the object surface by construction; projecting P into the first reference image with the reference's ground-truth pose and intrinsics gives the reference correspondence used in step (3).

Serialized Target. The language target serializes the four z-CoT steps as text: the normalized center projection (u_n, v_n) in $[0, 1000]^2$, the surface depth z_{surf} in millimeters, the reference correspondence, the signed offset $\Delta z_{\text{surf}\to\text{center}} = t_z^* - z_{\text{surf}}$, and the final center depth t_z^* inside <origin>.

Inference Inputs. The object mesh, the ground-truth segmentation mask, the rendered visibility, and the ground-truth query pose are used only for constructing these targets. At inference, PoseVLM uses only the RGB-D observations and the posed reference as input.

I STEP-WISE ACCURACY AND CONSISTENCY OF Z-COT

This analysis checks the accuracy of each generated z-CoT step and its relationship with the final pose. We use the final checkpoint on the full single-reference LINEMOD test set with the same greedy decoding as the main experiments. The model generates the entire chain without ground-truth intermediate steps. Evaluation targets follow Appendix H; surface depth and offset are scored against the same selected surface point, with $\Delta z^* = t_z^* - z_{\text{surf}}^*$. Ground-truth poses, masks, and CAD geometry are used only to construct these evaluation targets.

Step accuracy. z-CoT recovers both terms of the center-depth decomposition with millimeter-scale mean absolute errors: 3.4 mm for visible surface depth and 2.8 mm for the signed surface-to-center offset (Table 10). The center projection has a mean Euclidean error of 3.7 on the normalized $[0, 1000]$ coordinate scale. Together, these results show that the generated intermediate quantities closely match their geometric targets during inference.

Correlation with final pose accuracy. More accurate intermediate depth estimates are associated with more accurate final poses. The offset error has the strongest Spearman correlation with ADD/d ($\rho = 0.74$), followed by surface depth (0.66) and center projection (0.38). This pattern supports the relevance of the surface-to-center correction to final pose accuracy. Correlations are computed within each object and averaged with equal weights over objects with a defined correlation.

Table 10: **z-CoT step accuracy and its correlation with final pose accuracy on LINEMOD.** Mean errors of each generated step (left) and Spearman ρ between each step's error and the final ADD/d (right). Positive ρ means larger intermediate errors accompany larger pose errors.

Quantity	Mean Error ↓	Spearman ρ with ADD/d
Center projection (normalized units)	3.7	0.38
Surface depth (mm)	3.4	0.66
Surface-to-center offset (mm)	2.8	0.74

J BASELINE LATENCY SOURCES

The baseline latencies in Figure 6 are our measurements on the same A100 as PoseVLM; Table 11 also lists the values reported by the corresponding papers on their original hardware.

Table 11: Single-reference latency on LINEMOD; for OneViewAll, N denotes the number of pose hypotheses.

Method	Latency (s)	Source
PoseVLM (8B)	0.07	ours, A100
OneViewAll (N=12)	0.05	ours, A100
OneViewAll (N=78)	0.22	ours, A100
One2Any	0.06	ours, A100
SinRef-6D	0.74	ours, A100
FoundationPose	0.79	ours, A100
OneViewAll (N=12)	0.08	OneViewAll, Tab. 1 (RTX 4090)
OneViewAll (N=78)	0.38	OneViewAll, Tab. 8 (RTX 4090)
One2Any	0.09	One2Any, Tab. 5 (RTX 4090)
SinRef-6D	1.26	SinRef-6D, Tab. 7 (RTX 4090)
FoundationPose	1.30	FoundationPose, Sec. 5 (RTX 3090)

K DETAILED QUANTITATIVE RESULTS

This section reports the object-level counterparts of the aggregate comparisons in the main paper.

Table 12: Single-reference LINEMOD ADD-0.1d (%). GT-t denotes ground-truth translation; for OneViewAll, N denotes the number of pose hypotheses per query.

Method	Input	Reference	ape	bench.	cam	can	cat	drill.	duck	egg.	glue	hole.	iron	lamp	phone	Mean
NOPE (Nguyen et al., 2024)	RGB	1 + GT-t	2.0	4.5	2.5	2.2	0.7	4.7	0.5	100.0	79.4	2.9	4.5	4.2	3.9	16.3
Oryon (Corsetti et al., 2024)	RGB-D	1 real	1.2	1.3	3.9	0.8	12.7	8.5	0.8	63.2	18.4	1.6	0.6	2.9	11.7	9.8
One2Any (Liu et al., 2025a)	RGB-D	1 real	33.1	15.7	72.7	37.0	66.2	68.2	35.8	100.0	99.9	42.0	28.2	31.9	53.2	52.6
CoordAR (Zuo et al., 2026)	RGB-D	1 real	45.6	76.9	70.7	77.3	88.1	96.5	50.2	97.0	99.8	67.5	52.7	91.4	61.2	75.0
SinRef-6D (Liu et al., 2026)	RGB-D	1 rendered	85.7	99.3	73.2	98.3	93.0	98.7	66.6	98.5	99.1	74.6	90.9	97.6	97.4	90.3
OneViewAll (N=78) (Luo et al., 2026)	RGB-D	1 real	60.1	100.0	88.0	95.7	96.9	99.8	97.1	88.9	94.9	99.4	99.9	99.4	82.5	92.5
PoseVLM	RGB-D	1 real	**86.8**	96.5	**91.0**	97.8	**97.2**	97.4	96.3	**100.0**	99.9	92.6	91.7	92.3	91.8	**94.7**

Table 13: Object-level 16-reference LINEMOD ADD-0.1d (%), excluding refinement variants.

Method	ape	bench.	cam	can	cat	drill.	duck	egg.	glue	hole.	iron	lamp	phone	Mean
LatentFusion (Park et al., 2020)	88.0	92.4	74.4	88.8	94.5	91.7	68.1	96.3	94.9	82.1	74.6	94.7	91.5	87.1
FS6D (He et al., 2022)	74.0	86.0	88.5	86.0	98.5	81.0	68.5	100.0	99.5	97.0	92.5	85.0	99.0	88.9
FoundationPose (Wen et al., 2024)	99.0	100.0	100.0	100.0	100.0	100.0	99.4	100.0	100.0	99.9	100.0	100.0	100.0	99.9
PoseVLM	**99.3**	**100.0**	**100.0**	**100.0**	**100.0**	98.3	**99.5**	**100.0**	**100.0**	99.1	**100.0**	99.8	99.4	99.6

Table 14: Object-level 16-reference YCB-Video results using ADD/ADD-S AUC (%). PREDATOR (Huang et al., 2021), LoFTR (Sun et al., 2021), FS6D (He et al., 2022), and FoundationPose (Wen et al., 2024) serve as baselines.

	PREDATOR		LoFTR		FS6D		FoundationPose		**PoseVLM**	
Object	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S
002_master_chef_can	17.4	73.0	50.6	87.2	36.8	92.6	91.3	96.9	**91.8**	**98.1**
003_cracker_box	8.3	41.7	25.5	71.8	24.5	83.9	96.2	97.5	**97.1**	**98.4**
004_sugar_box	15.3	53.7	13.4	63.9	43.9	95.1	87.2	97.5	**87.8**	**98.5**
005_tomato_soup_can	44.4	81.2	52.9	77.1	54.2	93.0	93.3	97.6	**93.7**	**98.4**
006_mustard_bottle	5.0	35.5	59.0	84.5	71.1	97.0	97.3	98.4	**98.0**	**99.1**
007_tuna_fish_can	34.2	78.2	55.7	72.6	53.9	94.5	73.7	97.7	**74.9**	**98.5**
008_pudding_box	24.2	73.5	68.1	86.5	79.6	94.9	97.0	98.5	**97.2**	**99.4**
009_gelatin_box	37.5	81.4	45.2	71.6	32.1	98.3	97.3	98.5	**97.4**	**99.3**
010_potted_meat_can	20.9	62.0	45.1	67.4	54.9	87.6	82.3	96.6	**83.4**	**97.8**
011_banana	9.9	57.7	1.6	24.2	69.1	94.0	95.4	98.1	**95.6**	**99.0**
019_pitcher_base	18.1	83.7	22.3	58.7	40.4	91.1	96.6	97.9	**96.6**	**99.1**
021_bleach_cleanser	48.1	88.3	16.7	36.9	44.1	89.4	93.3	97.4	**93.4**	**98.5**
024_bowl	17.4	73.2	1.4	32.7	0.9	74.7	89.7	94.9	**91.1**	**97.3**
025_mug	29.5	84.8	23.6	47.3	39.2	86.5	75.8	96.2	**78.3**	**97.4**
035_power_drill	12.3	60.6	1.3	18.8	19.8	73.0	96.3	98.0	96.0	**98.6**
036_wood_block	10.0	70.5	1.4	49.9	27.9	94.7	94.7	97.4	**94.9**	**98.5**
037_scissors	25.0	75.5	14.6	32.3	27.7	74.2	95.5	97.8	**96.1**	**98.9**
040_large_marker	38.9	81.8	8.4	20.7	74.2	97.4	96.5	98.6	96.3	**99.5**
051_large_clamp	34.4	83.0	11.2	24.1	34.7	82.7	92.7	96.9	92.4	**98.9**
052_extra_large_clamp	24.1	72.9	1.8	15.0	10.1	65.7	94.1	97.6	**94.2**	95.8
061_foam_brick	35.5	79.2	31.4	59.4	45.8	95.7	93.4	98.1	93.1	**99.1**
Mean	24.3	71.0	26.2	52.5	42.1	88.4	91.5	97.4	**92.3**	**98.5**

Against FoundationPose with 16 references, PoseVLM attains higher ADD on 16 of the 21 objects, equal ADD on one, and higher ADD-S on 20.

Table 15: Effect of reference count. All scores are percentages. The same checkpoint is evaluated with different numbers of real reference views; only the first reference has an input pose. Higher is better for all three metrics.

References	LINEMOD	YCB-Video	
N_{ref}	ADD-0.1d	ADD AUC	ADD-S AUC
1	94.7	87.6	95.8
2	96.2	89.2	96.1
4	98.0	90.5	97.2
8	99.3	91.8	98.1
16	99.6	92.3	98.5

Table 16: Sensitivity to reference-frame selection on LINEMOD under an exhaustive protocol: every frame of each object serves as the reference in turn, and scores are averaged over all query frames. Mean, minimum, and maximum are ADD-0.1d scores (%); standard deviations are in percentage points (pp). All methods are evaluated on the same query–reference pairs.

Method	Mean (%)	Std. (pp)	Min. (%)	Max. (%)
OneViewAll (Luo et al., 2026)	90.1	1.15	88.2	93.1
SinRef-6D (Liu et al., 2026)	88.9	1.24	86.1	92.3
PoseVLM	93.4	0.36	91.8	94.0

L Additional Analyses

We complement the main experiments with additional analyses: the effect of reference count, sensitivity to single-reference selection, and tracking error over time.

L.1 Effect of the Number of Reference Views

Experimental protocol. We vary the number of reference views as $N_{\text{ref}} \in \{1, 2, 4, 8, 16\}$ on LINEMOD and YCB-Video. Following the main evaluation, we report ADD-0.1d on LINEMOD and ADD/ADD-S AUC on YCB-Video. We construct reference subsets from the origin 16-reference set used in the main experiments, following the sampling rule: keep the first reference fixed and uniformly sample the next $N_{\text{ref}} - 1$ references in the order they were originally posed.

Results. Table 15 shows that accuracy rises as more reference views are provided, with diminishing returns as the reference set grows. This matches the intuition that early views add complementary evidence while later views contribute little new information; the LINEMOD score approaches the metric ceiling at 16 references. Notably, the same predictor exploits this extra evidence without changing its parameters or introducing instance reconstruction.

L.2 Sensitivity to Single-Reference Selection

Experimental protocol. We evaluate how choosing a different reference frame affects single-reference accuracy on LINEMOD. For each object, we exhaustively traverse all frames of the sequence as references: every query frame is evaluated once with each of the other frames serving as the reference, yielding all valid query–reference pairs. PoseVLM is evaluated on all pairs; the compared baselines (OneViewAll (Luo et al., 2026), SinRef-6D (Liu et al., 2026)) are evaluated under the same protocol. Let $a_{o,r}$ denote the ADD-$0.1d$ score of object o when frame r serves as the reference, averaged over all query frames $q \neq r$, using the same symmetry handling as the main LINEMOD evaluation. For each method, we report the mean, sample standard deviation, and minimum and maximum of the pooled $\{a_{o,r}\}$ over all objects and reference frames.

Results. As illustrated in Table 16, PoseVLM is less sensitive to the choice of reference frame than the compared baselines, with a smaller standard deviation and a higher minimum score.

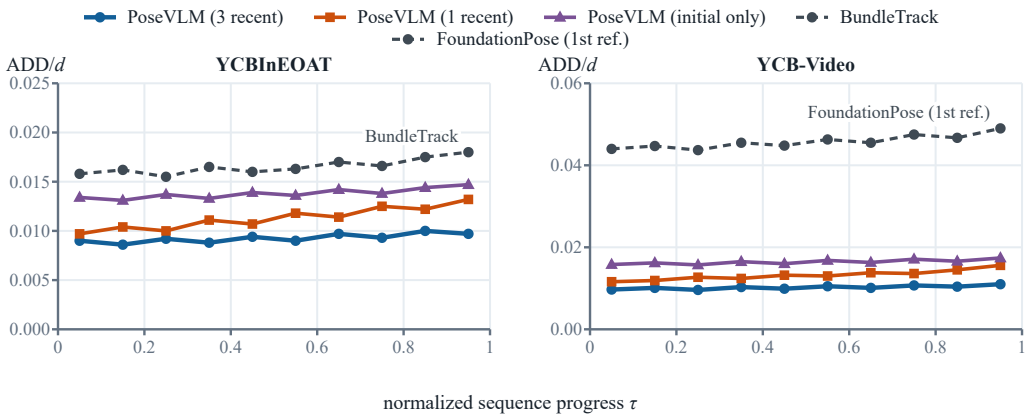

Figure 10: Tracking error over normalized sequence progress on YCBInEOAT and YCB-Video. Curves show diameter-normalized ADD error for the three reference-history configurations of PoseVLM and the compared baselines. Values are averaged within temporal bins and then equally across target-object tracks. All methods receive the ground-truth initialization frame and are evaluated without re-initialization. Lower is better.

L.3 Tracking Drift

Protocol and metrics. We examine the temporal behavior, especially the drift over time, of the tracking setting in Table 4 of the main paper on YCBInEOAT and YCB-Video. Predictions are produced causally: only the first frame receives the ground-truth pose, and subsequent ground-truth poses are used only to compute evaluation metrics. The compared tracking baselines (BundleTrack (Wen & Bekris, 2021) on YCBInEOAT and FoundationPose (Wen et al., 2024) on YCB-Video) are evaluated under the same protocol. For every post-initialization frame, we record the diameter-normalized ADD error $e_{s,t} = \mathrm{ADD}(\hat{T}_{s,t}, T^*_{s,t})/d_s$, where $\mathrm{ADD}(\cdot,\cdot)$ is the ADD point distance used by the main evaluation, $\hat{T}_{s,t}$ and $T^*_{s,t}$ are the predicted and ground-truth poses of track s at frame t, and d_s is the object diameter; lower is better. To compare tracks of different lengths, we normalize temporal progress as $\tau_{s,t} = (t-1)/(T_s-1)$, divide $(0,1]$ into ten equal bins, and average the frames of each track within a bin before averaging across tracks with equal weight, so that longer tracks do not dominate:

$$\bar{e}_{s,b} = \frac{1}{|\mathcal{I}_{s,b}|} \sum_{t \in \mathcal{I}_{s,b}} e_{s,t}, \qquad E_b = \frac{1}{|\mathcal{S}_b|} \sum_{s \in \mathcal{S}_b} \bar{e}_{s,b}, \tag{6}$$

where $\mathcal{I}_{s,b}$ contains the post-initialization frames of track s with $\tau_{s,t} \in ((b-1)/10, b/10]$, and $\mathcal{S}_b$ contains the tracks with at least one frame in that bin. All evaluated tracks and post-initialization frames are retained, including large-error predictions. We summarize early and late behavior by the first and last 20% of normalized progress: E_{early} and E_{late} average bins 1–2 and 9–10, and $\Delta E = E_{\mathrm{late}} - E_{\mathrm{early}}$. A positive ΔE indicates higher error toward the end of the tracks; since viewpoint and visibility also change over time, it is a descriptive measure of temporal error change, not an isolated estimate of drift.

Results. Figure 10 reports the temporal error curves, and Table 17 summarizes their early and late stages. With three recent predictions, PoseVLM's error stays at the lowest level of all curves and remains nearly flat across sequence progress: the early-to-late change in mean ADD/d is $+0.0011$ on YCBInEOAT and $+0.0008$ on YCB-Video. The three history configurations separate cleanly. With the initial reference only, errors start higher because a fixed reference cannot follow viewpoint and visibility changes; adding one recent prediction lowers the overall level but accumulates the steepest upward drift among the three, consistent with error feedback from recursive inference; three recent predictions absorb such single-frame errors and show both the lowest level and the smallest accumulation. Under the same protocol, both baselines start above and stay above every PoseVLM configuration: BundleTrack rises from 0.0160 to 0.0178 on YCBInEOAT, and FoundationPose remains far above all curves on YCB-Video (0.044–0.049 vs. at most 0.017). These measurements complement the aggregate tracking scores by revealing how error evolves within the evaluated

Table 17: Early-to-late tracking error. Early and late refer to the first and last 20% of normalized post-initialization sequence progress. Errors are normalized by object diameter, and $\Delta = \mathrm{Late} - \mathrm{Early}$. All methods are evaluated under the same ground-truth initialization without re-initialization.

Dataset	Method / History	Early	Late	Δ
YCBInEOAT	PoseVLM (initial only)	0.0133	0.0146	+0.0013
	PoseVLM (1 recent)	0.0100	0.0127	+0.0026
	PoseVLM (3 recent)	0.0088	0.0099	+0.0011
	BundleTrack	0.0160	0.0178	+0.0018
YCB-Video	PoseVLM (initial only)	0.0160	0.0170	+0.0010
	PoseVLM (1 recent)	0.0118	0.0151	+0.0033
	PoseVLM (3 recent)	0.0099	0.0107	+0.0008
	FoundationPose	0.0444	0.0479	+0.0035

sequences; they characterize finite-sequence stability rather than guaranteeing drift-free operation or recovery after arbitrarily long tracking loss.