Planning Where to Look: Foundation-Model-Guided Active View Planning for Reliable Manipulation

Author Names Omitted for Anonymous Review.

***Abstract*—Foundation models are increasingly used to propose robot actions, but reliable deployment also requires planning actions that improve the robot's observations. We study this problem through *ActivePose*, a closed-loop framework that plans where a robot-mounted camera should move to resolve 6-DoF object-pose ambiguity and maintain observability during manipulation. A vision–language model (VLM) is grounded with entropy-ranked CAD renderings that demonstrate object- and viewpoint-specific ambiguity. At runtime, the system evaluates the current view, imagines inverse-kinematics-feasible candidate views, and ranks them by fusing VLM ambiguity with pose-hypothesis entropy. The selected next-best view is executed in a receding-horizon loop. During manipulation, a demonstration-trained diffusion policy produces short-horizon camera trajectories that preserve visibility under object motion and occlusion. Experiments in simulation and on real dual-arm hardware show that the proposed sensing planner substantially improves pose estimation, tracking, and peg-in-hole assembly. These results illustrate how foundation models can complement geometric planning without replacing feasibility checks, structured uncertainty, or closed-loop verification.**

***Index Terms*—Robot planning, active perception, vision–language models, next-best-view planning, 6D object pose estimation, robotic manipulation.**

I. Introduction

Robot planning is often formulated around task actions such as reaching, grasping, assembly, or navigation. In practice, however, execution can fail because a planner acts on an incorrect or weakly observed state. This is especially problematic for symmetric and textureless objects, where a single RGB-D image may support multiple plausible 6D poses, and during manipulation, where self-occlusion or object motion can cause tracking loss. A reliable robot therefore needs to plan not only *what to do*, but also *where to look before and while acting*.

Foundation models offer a useful semantic prior for this decision. Vision–language models can reason about qualitative properties of observations and have been incorporated into task planning and embodied control [3, 7, 18]. Yet generic VLM judgments are not necessarily calibrated to the symmetries of a particular CAD model, nor do they guarantee robot feasibility. Conversely, geometric uncertainty can quantify pose ambiguity but may be brittle under rendering-to-real gaps and partial occlusion. We investigate a hybrid planning architecture in which a foundation model supplies a semantic ambiguity estimate, geometry supplies a structured uncertainty signal, and classical inverse kinematics constrains the action set.

We present *ActivePose* through the lens of foundation-model-guided robot planning. Its first module is a repeated one-step sensing planner: detect whether the current observation is ambiguous, generate local camera candidates, reject kinematically infeasible candidates, score imagined observations, execute the best view, and replan from the new real observation. Its second module is a learned receding-horizon planner that generates camera trajectories from demonstrations to maintain visibility during manipulation.

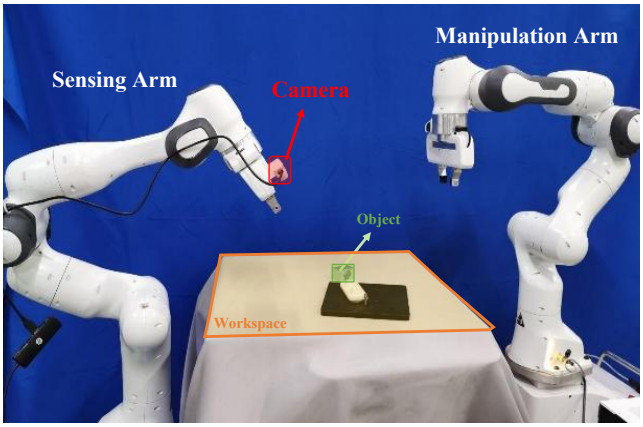

Fig. 1. Dual-arm setup. A sensing arm carries a wrist-mounted RGB-D camera, while a manipulation arm interacts with the object. The sensing planner changes viewpoint when the current observation is ambiguous or likely to lose visibility.

Our contributions are:

- a foundation-model-guided active-view planner grounded by entropy-ranked CAD observations rather than text-only prompts;
- a hybrid decision rule combining semantic ambiguity, pose-hypothesis entropy, and inverse-kinematics feasibility in a closed loop;
- a learned receding-horizon sensing planner and real-robot evidence that information-gathering actions improve pose reliability and downstream assembly;
- an analysis of latency and missing constraints that clarifies the role of foundation models inside practical robot planning systems.

II. Related Work

A. Foundation Models for Robot Planning

Large multimodal models have been used to translate language and perception into task plans, executable programs, or visuomotor actions [3, 7, 18, 5]. Our setting is complementary: the VLM does not generate a long-horizon manipulation plan. Instead, it evaluates whether an imagined sensor observation is likely to be ambiguous, while geometry and robot kinematics determine the feasible sensing action.

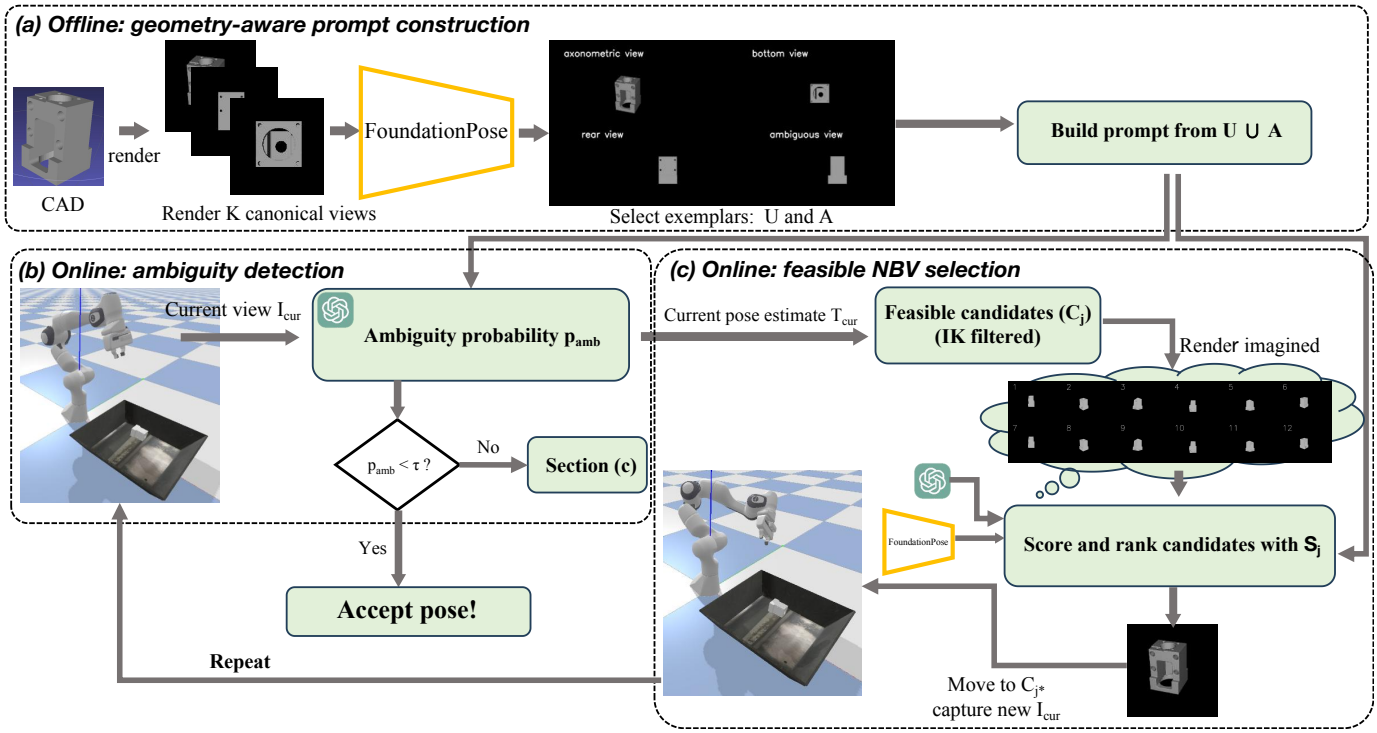

Fig. 2. **Foundation-model-guided active view planning.** Offline, entropy-ranked CAD views ground the VLM in object-specific pose ambiguity. Online, ambiguity triggers a planning loop: generate IK-feasible candidates, render imagined observations, fuse semantic ambiguity and pose-hypothesis entropy, execute the best next view, and replan from the resulting real observation.

B. Active Perception and Next-Best-View Planning

Single-view 6D pose estimators have progressed from object-specific supervised methods [13, 23, 6, 12, 4, 14] to zero-shot systems that use CAD models for novel objects [10, 20, 8]. Active pose-estimation methods select observations that reduce uncertainty using reinforcement learning, geometric objectives, or learned next-view predictors [15, 9, 22]. Active-Pose grounds a general VLM with object-specific geometric examples and combines its semantic score with uncertainty from a CAD-based pose estimator.

C. Learned Receding-Horizon Sensing

Diffusion policies generate smooth, multimodal action sequences in a receding-horizon loop [21, 2]. Unlike work that uses diffusion models primarily to generate manipulation actions, we use one as an active sensing planner. It predicts wrist-camera trajectories that preserve target visibility and recover after temporary occlusion.

III. Planning Problem and System Overview

We formulate active 6D pose estimation and tracking as a closed-loop sensing-planning problem. Given a CAD model, an initial camera pose, and an RGB-D stream, the robot must estimate the object pose while deciding whether another observation is worth the motion and sensing cost. During initialization, high ambiguity triggers a next-best-view action selected from imagined, kinematically feasible observations. During manipulation, a learned sensing policy repeatedly plans camera motion to preserve visibility and avoid pose loss. The resulting perceive–evaluate–plan–execute loop is receding horizon: imagined views are used only to select an action, and every executed action is verified through a new real observation.

IV. Hybrid Active View Planning

Our approach contains two coupled modules. The first performs geometry-grounded ambiguity assessment and next-best-view selection (Algorithm 1); the second learns a receding-horizon sensing policy that maintains observability during manipulation (Algorithm 2). Together they implement a perceive–evaluate–plan–execute loop.

A. Active Pose Estimation

Our active pose estimation pipeline (see Fig. 2) comprises two stages: (i) *offline* geometry-aware prompt construction and (ii) *online* ambiguity detection with feasible NBV selection.

Offline: We pre-render K canonical views $\{R_k\}_{k=1}^{K}$ by sampling camera poses on a tabletop-facing hemisphere around the object (fixed standoff distance d, camera optical axis pointing to the object center), and reuse these renders to construct a geometry-aware prompt that remains fixed for all online queries of the same object. These canonical views are used only for prompt construction and do not need to be kinematically feasible for the robot. For each rendered view R_k, we invoke FoundationPose [20] to obtain the top-N *6D* pose hypotheses $\{\widehat{T}_i\}_{i=1}^{N}$ with associated confidence weights

Algorithm 1 Active Pose Estimation

Require: CAD model; ambiguity threshold τ; fusion weight λ; NBV budget L; candidate-view generator $\mathcal{G}(\cdot)$ with IK filtering

Ensure: final 6D object pose in camera frame ${}^{c}T_o$

Offline:
Render K canonical views $\{R_k\}_{k=1}^{K}$
Compute normalized pose entropies $\{\bar{H}_k\}_{k=1}^{K}$ from FoundationPose hypotheses (Eq. (1))
$U \leftarrow K_u$ lowest-entropy views; $A \leftarrow K_a$ highest-entropy views
prompt $\leftarrow$ BUILDPROMPT(U, A)
Online:
Acquire I_{cur} at current camera pose ${}^{b}T_c$
$T_{\text{cur}} \leftarrow \text{FoundationPose}(I_{\text{cur}})$
for $\ell = 0$ **to** L **do**
  $p_{\text{amb}} \leftarrow$ AMBIGUITYPROB$(I_{\text{cur}}, \text{prompt})$ (Eq. (2))
  if $p_{\text{amb}} < \tau$ **or** $\ell = L$ **then**
    break
  end if
  ${}^{b}T_o \leftarrow {}^{b}T_c \cdot T_{\text{cur}}$
  $\{C_j\}_{j=1}^{M} \leftarrow \mathcal{G}({}^{b}T_c, {}^{b}T_o)$
  for $j = 1, \ldots, M$ **do**
    $\hat{I}_j \leftarrow \text{Render}(C_j)$
    $p_{\text{amb},j} \leftarrow$ AMBIGUITYPROB$(\hat{I}_j, \text{prompt})$ (Eq. (2))
    $\bar{H}_j \leftarrow$ POSEENTROPY$(\hat{I}_j)$ (Eq. (1))
    $S_j \leftarrow \lambda \bar{H}_j + (1-\lambda) p_{\text{amb},j}$
  end for
  $j^* \leftarrow \arg\min_j S_j$
  Move camera to C_{j^*} and acquire new real image I_{cur}
  ${}^{b}T_c \leftarrow C_{j^*}$
  $T_{\text{cur}} \leftarrow \text{FoundationPose}(I_{\text{cur}})$
end for
${}^{c}T_o \leftarrow T_{\text{cur}}$

$\{w_i\}_{i=1}^{N}$. We normalize the weights to form a discrete distribution $p_i = \frac{w_i}{\sum_{j=1}^{N} w_j}$ and compute the *normalized* Shannon entropy

$$\bar{H}(R_k) = \frac{-\sum_{i=1}^{N} p_i \log p_i}{\log N} \in [0, 1]. \tag{1}$$

This serves as a geometry-driven uncertainty score that is comparable across views. We then select two small subsets: U as the K_u lowest-entropy *unambiguous* views and A as the K_a highest-entropy *ambiguous* views. The geometry-aware prompt is constructed by aggregating the images in $U \cup A$ with a fixed instruction template.

Online: Given the current observation I_{cur}, we estimate an initial pose $T_{\text{cur}} = \text{FoundationPose}(I_{\text{cur}})$ and compute an ambiguity probability p_{amb} using a fixed, per-object prompt (constructed offline) and constrained $\{\texttt{Yes}, \texttt{No}\}$ log-probabilities from the VLM:

$$\begin{aligned} p_{\text{amb}} &= P_{\text{VLM}}(\texttt{Yes} \mid I_{\text{cur}}, \text{prompt}) \\ &= \frac{\exp(\ell_{\text{Yes}})}{\exp(\ell_{\text{Yes}}) + \exp(\ell_{\text{No}})}. \end{aligned} \tag{2}$$

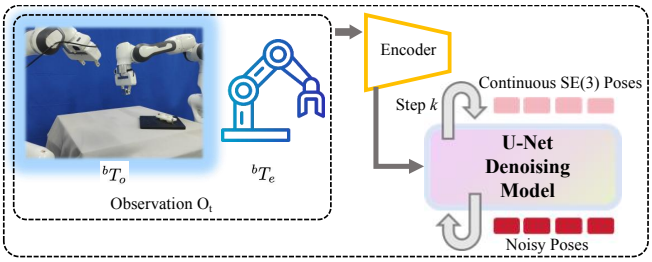

Fig. 3. **Active pose tracking.** The policy encodes the current observation O_t, denoises over K_d reverse-diffusion steps to generate a horizon of continuous SE(3) poses, and executes the last k_h poses in a receding-horizon loop.

If $p_{\text{amb}} < \tau$, we directly output T_{cur}; otherwise we iteratively select a feasible NBV within a small budget L. A larger τ makes disambiguation less frequent (only highly ambiguous views trigger NBV), while a smaller τ triggers NBV more aggressively. Given the current camera pose ${}^{b}T_c$ and the current pose estimate T_{cur}, the generator $\mathcal{G}$ samples M *local* candidate camera poses around the object and retains only IK-feasible candidates. For each feasible candidate C_j, we render an imagined view $\hat{I}_j$ and compute $p_{\text{amb},j} = P_{\text{VLM}}(\texttt{Yes} \mid \hat{I}_j, \text{prompt})$ with the same prompt. We score candidates by $S_j = \lambda\, \bar{H}(\hat{I}_j) + (1-\lambda)\, p_{\text{amb},j}$, then move the camera to the best-scoring candidate view, and acquire a new real observation. We repeat this process until the newly acquired image yields $p_{\text{amb}} < \tau$ or the budget L is exhausted (Algorithm 1); at termination, we return the current estimate as the final pose ${}^{c}T_o$.

B. Active Pose Tracking

Algorithm 2 Active Pose Tracking

Require: Object pose in camera frame ${}^{c}T_o$, initial end-effector pose ${}^{b}T_{e,1}$, camera pose in end-effector frame ${}^{e}T_c$, history length H, prediction horizon K, executed tail length k_h, total steps N.

Ensure: executed poses $\{{}^{b}T_{e,t}\}_{t=1}^{N}$

$t \leftarrow H$
while $t \leq N$ **do**
  Compute object pose in base frame: ${}^{b}T_o$ (Eq. (3))
  Collect history window: O_t
  Sample a future end-effector pose sequence by denoising:
  $$\{{}^{b}\widehat{T}_e^{\,t+1:t+K}\} \leftarrow \text{DP}(O_t)$$
  ▷ DP denotes the diffusion policy, see Eqs. (4)–(5)
  Select the **last** k_h **poses** of the predicted horizon K:
  $$\mathcal{T}_t \leftarrow \{{}^{b}\widehat{T}_e^{\,t+K-k_h+1:t+K}\}$$
  Time-parameterize and interpolate from current ${}^{b}T_e$ to $\mathcal{T}_t$, then execute the resulting continuous trajectory via servo control.
  Update ${}^{b}T_e$ to the last executed pose: ${}^{b}T_e \leftarrow {}^{b}\widehat{T}_e^{\,t+K}$
  $t \leftarrow t + k_h$
end while

After obtaining the final disambiguated pose ${}^{\mathcal{T}}T_o$ from active pose estimation, we formulate active tracking as a *conditional diffusion* problem to maintain continuous visibility (see Fig. 3). Because the camera is rigidly mounted on the sensing end-effector with a fixed extrinsic ${}^{e}T_c$, a predicted end-effector trajectory $\{{}^{b}T_{e,t}\}$ uniquely induces a camera trajectory $\{{}^{b}T_{c,t}\}$ via ${}^{b}T_c = {}^{b}T_e \cdot {}^{e}T_c$. For clarity, we learn and execute end-effector trajectories, which are equivalent to camera trajectories under this rigid mounting.

At each time step t, the pose estimator provides the object pose in the camera frame ${}^{c}T_o$. We convert it to the robot base frame using the current end-effector pose ${}^{b}T_e$ and the fixed camera extrinsic ${}^{e}T_c$:

$$ {}^{b}T_o = {}^{b}T_c \cdot {}^{c}T_o = {}^{b}T_e \cdot {}^{e}T_c \cdot {}^{c}T_o. \tag{3} $$

We then build the observation $O_t = \{{}^{b}T_o^{t-H+1:t},\ {}^{b}T_e^{t-H+1:t}\}$, containing the past H frames of (i) the object pose in the robot base frame ${}^{b}T_o$ and (ii) the end-effector pose in the base frame ${}^{b}T_e$. We condition the policy on a short history of poses rather than a single-frame estimate, which provides motion context and improves tracking robustness.

Specifically, starting from Gaussian noise $A_t^{K_d}$, the denoising network ε_θ performs K_d reverse-diffusion steps to transform $A_t^{K_d}$ into a noise-free action A_t^0:

$$ A_t^{k-1} = \alpha_k\Big(A_t^k - \gamma_k\, \varepsilon_\theta(O_t, A_t^k, k)\Big) + \sigma_k\, \mathcal{N}(0, I), \tag{4} $$

$$ \mathcal{L} = \mathbb{E}\Big[\big\|\epsilon^k - \varepsilon_\theta\big(O_t,\ \hat{\alpha}_k A_t^0 + \hat{\beta}_k \epsilon^k,\ k\big)\big\|^2\Big], \tag{5} $$

where $\alpha_k, \gamma_k, \sigma_k$ are parameters of the noise schedule at step k, ϵ^k is the injected noise, and $\hat{\alpha}_k, \hat{\beta}_k$ are the one-step forward diffusion coefficients [16]. Mean squared error (MSE) loss is employed as the objective function to supervise noise prediction (Eq. (5)). The rotation is represented as R6D [19], a continuous 6D representation for $SO(3)$ that avoids discontinuities and is empirically stable for generative modeling. At runtime, the diffusion policy outputs an absolute horizon $\{{}^{b}\widehat{T}_e^{t+1:t+K}\}$. Instead of executing a prefix, we execute the *last* k_h poses $\{{}^{b}\widehat{T}_e^{t+K-k_h+1:t+K}\}$, biasing the camera toward the terminal viewpoint that best preserves future visibility. We interpolate from the current ${}^{b}T_e$ to this segment for smooth, feasible motion and repeat in a receding-horizon loop.

V. Experiments

This section describes the experimental setups, implementation details, evaluation protocol, and quantitative results for active pose estimation and active pose tracking, followed by ablations, runtime analysis, and an engineering case study on peg-in-hole assembly.

A. Experimental Setups

Hardware/Software and Objects. For **active pose estimation**, we evaluate in both simulation and the real world. Simulation is implemented in PyBullet [11], which provides controlled rendering for NBV selection. For **active pose tracking**, we exclusively perform real-robot evaluations to capture realistic occlusions, dynamics, and sensing noise. All real-world experiments use two Franka Emika Panda 7-DOF arms: a *sensing arm* carries a wrist-mounted Intel RealSense D435 RGB-D camera (640×480 at 30 Hz), and a *manipulation arm* is equipped with a parallel-jaw gripper (see Fig. 1). As shown in Fig. 4, we select three objects from MP6D [1] for evaluation and an additional object for the peg-in-hole assembly task. All four objects exhibit strong viewpoint-induced ambiguities due to symmetry and textureless geometry.

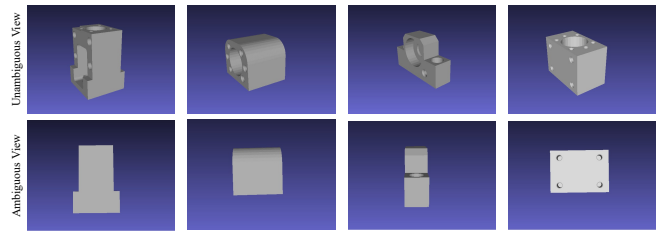

Fig. 4. CAD models of the experimental objects (Obj. 1–3 from MP6D; Obj. 4 for assembly) and an example ambiguous view.

Implementation details. *Active pose estimation:* We use GPT-4o as the VLM through an external API. The VLM operates on a cropped image (resized to 224×224) paired with a fixed, per-object geometry-aware prompt, while FoundationPose uses the full RGB-D observation. VLM inference is run deterministically: we set `temperature` = 0 (and `top_p` = 1.0) and compute the ambiguity probability from the returned {Yes, No} log-probabilities (Eq. (2)); all other decoding options are fixed across experiments. Offline prompting uses K=48 rendered views, from which we select K_u=K_a=2 exemplars; the ambiguity threshold is fixed to τ=0.5. When ambiguity is triggered, we render M=12 feasible candidate viewpoints and score them with fusion weight λ=0.6. We use fixed L=3 across all objects and scenarios.

Prompt template: For each object, the prompt consists of (i) a short instruction asking whether the pose is ambiguous and constraining the answer to {`Yes`, `No`}, (ii) K_u low-entropy exemplar renders labeled `No` and K_a high-entropy exemplar renders labeled `Yes`, and (iii) the query image.

Active pose tracking: We collect 10 expert demonstrations per task, each consisting of ∼200 frames of object poses ${}^{b}T_o$ and end-effector poses ${}^{b}T_e$ at 10 Hz, via teleoperation that keeps the target within the camera frustum. We train the diffusion policy with AdamW (batch size 40, learning rate 3×10^{-4}) for 2,000 epochs. We use fixed H=5, K=10, k_h=5 and N=200 across all objects and scenarios. All training and inference run on a single NVIDIA RTX 4090 GPU.

Estimation scenarios. We evaluate active pose estimation under two conditions:

1) *Random Placement:* each object is placed at a random position and orientation on the workspace.
2) *High-Entropy Placement:* each object is deliberately positioned at a precomputed high-entropy (ambiguous) viewpoint.

Estimation baselines.

- **Fixed-View:** run FoundationPose once at the initial camera pose (no active viewpoint selection).
- **Random-NBV:** upon ambiguity detection, randomly select one of the M feasible candidate viewpoints for a second FoundationPose estimate.
- **Entropy-NBV:** upon ambiguity detection, select NBV by minimizing the normalized hypothesis entropy $\bar{H}_j$ over the M feasible candidates (no VLM).
- **VLM-NBV:** upon ambiguity detection, select NBV by minimizing VLM ambiguity $p_{\mathrm{amb},j}$ over the M feasible candidates (no FoundationPose).

Estimation metric. We report *success rate* (SR), defined as the percentage of trials whose final estimated 6D pose has translation error below 5 mm and rotation error below 5°. Each method is tested for 100 trials per scenario.

Tracking scenarios. We evaluate all trackers under four challenging conditions:

1) *Long-range linear motion:* the object moves along a straight trajectory from point A to point B, where point B is randomized.
2) *Circular rotational motion:* the object rotates around a fixed point on the table, inducing large viewpoint changes.
3) *Temporary occlusion:* the object is briefly occluded (up to 1 s), requiring recovery upon reappearance.
4) *Random spatial motion:* the object undergoes unpredictable translational and rotational motions in 3D, including sharp turns and speed variations.

Tracking baselines.

- **Pose-Servo:** a classical pose-based visual servoing baseline. The wrist-mounted camera provides the current 6D object pose; a proportional controller maps pose error to end-effector velocity commands.
- **World-Camera:** a fixed, calibrated overhead camera continuously tracks the object's 6D pose via FoundationPose.

Tracking metric. We report *success rate* (SR) per trial: the fraction of runs that complete the scenario without any *pose-loss* event. We declare pose-loss if (i) FoundationPose returns no valid pose for more than 10 consecutive frames, or (ii) the object exits the camera FOV. Each method is evaluated in 50 independent trials per scenario.

B. Experimental Results

Active Pose Estimation Results. We evaluate two placement conditions: *Random Placement* and *High-Entropy Placement* (deliberately ambiguous starts). Unless otherwise stated, SR is averaged over four objects (Obj. 1–4).

Simulation. As reported in Table I, Fixed-View degrades sharply from Random Placement (60.0%) to High-Entropy Placement (20.0%), highlighting strong viewpoint-induced ambiguity. Random-NBV (28.3%) improves over Fixed-View (20.0%), confirming that acquiring an additional view can help, but remains unreliable without guided selection. Entropy-NBV performs competitively under Random Placement (64.3%) but drops under High-Entropy Placement (42.0%), suggesting that hypothesis entropy alone can be an imperfect proxy for selecting a truly disambiguating view. VLM-NBV performs worst among active baselines (41.0% / 34.8%), indicating that VLM ambiguity alone is insufficient for reliable NBV ranking over rendered candidates. In contrast, ActivePose (97.5% / 95.0%) achieves consistently high SR in both conditions, demonstrating robust disambiguation (Table I).

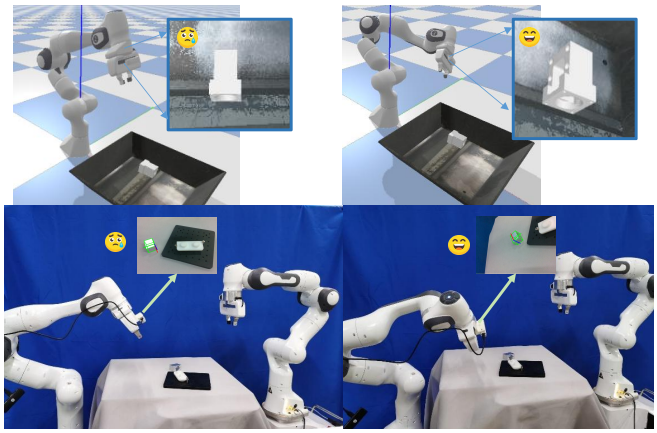

Fig. 5. **Example of active pose estimation.** An ambiguous initial view triggers NBV selection; moving to the selected viewpoint yields an unambiguous 6D pose estimate, shown in simulation (top) and real-robot trials (bottom).

Real world. Table I shows a similar trend: Fixed-View (21.8%) and Random-NBV (47.8%) drop substantially under High-Entropy Placement. Entropy-NBV yields moderate gains over Random-NBV in the High-Entropy setting (48.8% vs. 47.8%), while VLM-NBV remains unreliable in the real world (32.5% / 32.3%), consistent with the simulation results. In contrast, ActivePose achieves the best performance in both placements (92.5% on Random and 95.0% on High-Entropy), indicating robust disambiguation. The lower SR in real-world trials compared to simulation is likely due to real sensing imperfections, including sensor noise and the synthetic-to-real gap between rendered and captured observations. Figure 5 shows an example of active pose estimation, where an ambiguous view triggers NBV selection and a second view resolves the ambiguity.

Active Pose Tracking Results. Table II reports SR in 50 trials per scenario. Across all four motion scenarios, our tracker consistently outperforms Pose-Servo and World-Camera by large margins. Pose-Servo frequently fails due to reachability limitations under large viewpoint changes, while World-Camera fails when the object exits its fixed FOV. Unlike pose-based servoing, which enforces strict instantaneous pose-error tracking and can drive the arm into infeasible motions, our learned policy generates smooth, anticipatory camera motions that prioritize maintaining visibility and reacquiring the target after occlusion. Figure 6 illustrates an example of active pose tracking in our dual-arm setup.

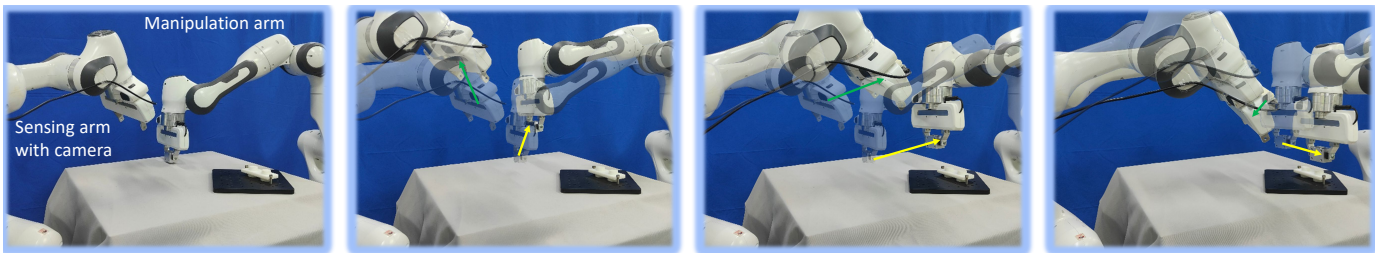

Fig. 6. **Example of active pose tracking.** While the manipulation arm manipulates the object, the sensing arm with a wrist-mounted RGB-D camera actively follows to keep the target in view.

TABLE I
ACTIVE POSE ESTIMATION IN SIMULATION AND ON REAL ROBOTS.

Method	Simulation										Real Robots									
	Random Placement					High-Entropy Placement					Random Placement					High-Entropy Placement				
	Obj 1	Obj 2	Obj 3	Obj 4	SR	Obj 1	Obj 2	Obj 3	Obj 4	SR	Obj 1	Obj 2	Obj 3	Obj 4	SR	Obj 1	Obj 2	Obj 3	Obj 4	SR
Fixed-View	70.0%	50.0%	50.0%	70.0%	60.0%	40.0%	10.0%	20.0%	10.0%	20.0%	68.0%	47.0%	42.0%	54.0%	52.8%	26.0%	21.0%	31.0%	9.0%	21.8%
Random-NBV	73.0%	51.0%	66.0%	70.0%	65.0%	41.0%	21.0%	34.0%	17.0%	28.3%	71.0%	52.0%	43.0%	64.0%	57.5%	55.0%	41.0%	67.0%	28.0%	47.8%
Entropy-NBV	65.0%	57.0%	61.0%	74.0%	64.3%	52.0%	38.0%	41.0%	37.0%	42.0%	73.0%	51.0%	50.0%	57.0%	57.8%	60.0%	46.0%	63.0%	26.0%	48.8%
VLM-NBV	67.0%	53.0%	25.0%	19.0%	41.0%	46.0%	34.0%	27.0%	32.0%	34.8%	37.0%	29.0%	23.0%	41.0%	32.5%	31.0%	37.0%	51.0%	10.0%	32.3%
Ours	**100.0%**	**100.0%**	**100.0%**	**90.0%**	**97.5%**	**100.0%**	**90.0%**	**100.0%**	**90.0%**	**95.0%**	**90.0%**	**90.0%**	**100.0%**	**90.0%**	**92.5%**	**100.0%**	**90.0%**	**100.0%**	**90.0%**	**95.0%**

TABLE II
ACTIVE POSE TRACKING ON REAL ROBOTS.

Method	Long-range Linear Motion					Circular Rotational Motion					Temporary Occlusion					Random Spatial Motion				
	Obj 1	Obj 2	Obj 3	Obj 4	SR	Obj 1	Obj 2	Obj 3	Obj 4	SR	Obj 1	Obj 2	Obj 3	Obj 4	SR	Obj 1	Obj 2	Obj 3	Obj 4	SR
Pose-Servo	65.0%	70.0%	60.0%	50.0%	61.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	5.0%	0.0%	1.3%	50.0%	45.0%	50.0%	55.0%	50.0%
World-Camera	50.0%	45.0%	65.0%	35.0%	48.8%	65.0%	60.0%	75.0%	50.0%	62.5%	35.0%	15.0%	15.0%	5.0%	17.5%	70.0%	55.0%	65.0%	45.0%	58.8%
Ours	**85.0%**	**90.0%**	**80.0%**	**95.0%**	**87.5%**	**80.0%**	**90.0%**	**95.0%**	**100.0%**	**91.3%**	**50.0%**	**45.0%**	**65.0%**	**50.0%**	**52.5%**	**70.0%**	**75.0%**	**65.0%**	**80.0%**	**72.5%**

TABLE III
ABLATION OF ACTIVE POSE ESTIMATION.

Method	Sim / Random		Sim / High-Entropy		Real / Random		Real / High-Entropy	
	SR	#NBV	SR	#NBV	SR	#NBV	SR	#NBV
Ours (entropy-ranked exemplars, λ=0.6)	**97.5%**	**0.60**	**95.0%**	**0.90**	**92.5%**	**0.60**	**95.0%**	**1.00**
Prompt grounding (exemplars)								
Text-only prompt (no exemplars)	73.0%	1.60	61.0%	1.30	70.5%	1.65	53.0%	2.00
Random exemplars (same count as K_u+K_a)	77.5%	1.10	68.0%	1.00	78.5%	0.90	70.5%	1.10
Fusion sensitivity (sweep λ)								
λ=0 (VLM-only)	41.0%	2.90	34.8%	3.00	32.5%	2.80	32.3%	3.00
λ=0.3	47.4%	2.70	36.8%	2.60	31.6%	2.60	25.4%	2.50
λ=0.9	61.7%	1.30	46.0%	1.40	65.5%	1.40	50.0%	1.70
λ=1 (entropy-only)	64.3%	1.40	42.0%	1.60	57.8%	1.50	48.8%	1.30

C. Ablation Studies

To isolate the specific value of using a VLM, to examine whether the VLM ambiguity score complements the FoundationPose entropy signal, and to assess the choice of the fusion weight λ, we conduct ablations on the active pose estimation module.

We ablate active pose estimation on the same four objects under both placement conditions, using the same SR metric (5 mm / 5°) and default settings (K=48, K_u=K_a=2, M=12, τ=0.5, L=3). In addition to SR, we report #NBV: the average number of executed NBV actions (additional viewpoints beyond the initial view) per trial. A lower #NBV is better.

Prompt grounding. Table III shows that entropy-ranked exemplars substantially outperform a text-only prompt and random exemplars, improving both SR and #NBV. Random exemplars keep the same exemplar count and labels as the full method, but randomly sample the low-/high-entropy exemplars from their respective pools instead of taking the entropy extremes. Notably, text-only prompting frequently exhausts the NBV budget (#NBV $\approx$ 1.3–2.0) while still yielding low SR, indicating that exemplars are critical for calibrating the VLM to *geometric* ambiguity.

Fusion sensitivity. Sweeping λ confirms that intermediate fusion (λ=0.6) provides the best SR–#NBV trade-off. At λ=0 (VLM-only), the policy often hits the budget limit (#NBV $\approx$ 2.5–3.0) yet remains inaccurate, showing that VLM ambiguity alone is insufficient for reliable NBV ranking. At λ=1 (entropy-only), #NBV decreases but SR remains substantially below the fused setting, suggesting that hypothesis entropy alone can be misled by rendering-to-real discrepancies and

TABLE IV
RUNTIME BREAKDOWN OF ACTIVE POSE ESTIMATION.

Component	Time (ms)
FoundationPose (single view, per image)	60
Render M=12 imagined views (total)	80
VLM query (per call, mean $\pm$ std)	600 ± 200
NBV evaluation (12 VLM calls + scoring)	7,500
End-to-end NBV cycle (worst case, incl. motion)	11,000

TABLE V
PEG-IN-HOLE ASSEMBLY.

Method	SR
Fixed-View + World-Camera	40%
Fixed-View + Pose-Servo	50%
Random-NBV + Pose-Servo	70%
Ours	90%

partial occlusions. Overall, the results support that VLM ambiguity and hypothesis entropy provide complementary cues for selecting disambiguating viewpoints.

D. Runtime and Deployment Considerations

The active estimation loop runs FoundationPose in 60 ms and renders 12 candidate views in 80 ms, but sequential external VLM calls dominate latency: candidate evaluation takes about 7.5 s and a worst-case NBV cycle, including motion, takes about 11 s (Table IV). The VLM is queried only at initialization or recovery, not in the high-frequency tracking loop. Nevertheless, local models, batched candidate evaluation, and calibrated early stopping are important steps toward deployment.

E. Engineering Case Study: Peg-in-Hole Assembly

Beyond isolated estimation and tracking, we evaluate a peg-in-hole assembly task as an *engineering case study* to demonstrate the practical utility of ActivePose in a closed-loop manipulation pipeline. In each run, the robot grasps the object from the tabletop and inserts it into a fixed socket whose position is randomized. We use the assembly policy from [17], which takes as input the relative SE(3) pose between the grasped object and the socket and outputs short-horizon end-effector motions to complete the assembly task. ActivePose provides this relative pose online by combining (i) active disambiguation at grasp initialization and (ii) active tracking during insertion to maintain visibility under motion and occlusions. This setting jointly tests reliable initialization (resolving grasp-time ambiguity) and sustained visibility during insertion.

Baselines and metric. We compare against three baselines: **Fixed-View + World-Camera** (single estimate from a fixed world camera), **Fixed-View + Pose-Servo** (single estimate at grasp followed by pose-based visual servoing during insertion), and **Random-NBV + Pose-Servo** (one random feasible NBV at grasp for a second estimate, then pose-servo during insertion). We report success rate (SR) as the fraction of trials that complete the insertion (20 trials per method).

Results and discussion. Table V shows that ActivePose achieves the best performance (90%). Fixed-View + World-Camera often fails when the randomized socket falls outside the fixed FOV (40%), while Fixed-View + Pose-Servo is limited by reachability and pose-loss during insertion (50%). Random-NBV + Pose-Servo reduces grasp-time ambiguity and improves SR to 70%, but still suffers from pose-loss under a fixed-view servo controller. By actively disambiguating at grasp and tracking during insertion, ActivePose maintains visibility and provides reliable relative poses for the downstream insertion.

VI. PLANNING PERSPECTIVE

ActivePose demonstrates one role for foundation models in robot planning: evaluating imagined observations inside a model-based, feasibility-filtered loop. The VLM supplies a semantic heuristic rather than a motion plan or a formal guarantee. Entropy-ranked CAD examples ground the heuristic in object-specific geometry, while inverse kinematics prevents obviously infeasible sensing actions. Closed-loop re-observation is equally important: rendered candidate views guide action selection, but the robot reruns perception on a real observation before committing to downstream manipulation. The ablations show that the VLM should complement rather than replace structured geometric objectives.

The system also connects three planning regimes. First, initialization is a one-step information-gathering planner over a discrete, IK-filtered candidate set. Second, repeated NBV selection forms a receding-horizon loop over the perceptual belief induced by pose hypotheses. Third, active tracking uses a learned short-horizon trajectory generator to preserve observability under object motion. This combination suggests a broader planning architecture in which foundation models evaluate semantically meaningful consequences, geometric models supply state and uncertainty, and classical constraints define executable actions.

VII. LIMITATIONS AND FUTURE PLANNING DIRECTIONS

Several gaps remain before this becomes a general active-sensing planner. First, the candidate generator samples a finite local set and filters only inverse-kinematics feasibility; it does not jointly optimize collision risk, path length, energy, sensing time, or downstream task value. Second, the ambiguity score is not probabilistically calibrated, and external VLM calls dominate the worst-case planning cycle. Third, the method requires a CAD model and object-specific prompt construction. Fourth, the learned tracking planner is trained from ten demonstrations per task and has not been evaluated for broad cross-object or cross-task transfer. Finally, experiments cover four objects and one assembly case study. Future work should formulate view selection as constrained belief-space planning, distill or batch foundation-model evaluation, incorporate collision-aware trajectory optimization, and couple information gain directly with task-level costs and safety constraints.

VIII. Conclusion

We presented ActivePose as a hybrid sensing planner that decides when and where a robot should look before and during manipulation. Geometry-grounded VLM evaluation, pose-hypothesis entropy, kinematic feasibility, and closed-loop replanning jointly produce reliable information-gathering actions. A diffusion policy extends this planning perspective to camera trajectories that maintain visibility during manipulation. The real-robot results suggest that foundation models can add useful semantic structure to robot planning, provided they operate within explicit geometric constraints and are verified through new observations after execution.

References

[1] Long Chen, Han Yang, Chenrui Wu, and Shiqing Wu. Mp6d: An rgb-d dataset for metal parts' 6d pose estimation. *IEEE Robotics and Automation Letters*, 7(3): 5912–5919, 2022. doi: 10.1109/LRA.2022.3154807.

[2] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024. URL https://arxiv.org/abs/2303.04137.

[3] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model, 2023. URL https://arxiv.org/abs/2303.03378.

[4] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation, 2020. URL https://arxiv.org/abs/1911.04231.

[5] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

[6] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7678–7687, 2019.

[7] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control, 2023. URL https://arxiv.org/abs/2209.07753.

[8] 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*, pages 27906–27916, 2024.

[9] Tomoki Mizuno, Kazuya Yabashi, and Tsuyoshi Tasaki. Object pose estimation by camera arm control based on the next viewpoint estimation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 9482–9487. IEEE, October 2024. doi: 10.1109/iros58592.2024.10801633. URL http://dx.doi.org/10.1109/IROS58592.2024.10801633.

[10] 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*, pages 9903–9913, 2024.

[11] Jacopo Panerati, Hehui Zheng, SiQi Zhou, James Xu, Amanda Prorok, and Angela P. Schoellig. Learning to fly – a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control, 2021. URL https://arxiv.org/abs/2103.02142.

[12] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7668–7677, 2019.

[13] 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*, pages 4561–4570, 2019.

[14] Ivan Shugurov, Sergey Zakharov, and Slobodan Ilic. Dpodv2: Dense correspondence-based 6 dof pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7417–7435, 2022. doi: 10.1109/TPAMI.2021.3118833.

[15] Juil Sock, Guillermo Garcia-Hernando, and Tae-Kyun Kim. Active 6d multi-object pose estimation in cluttered scenarios with deep reinforcement learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10564–10571, 2020. doi: 10.1109/IROS45743.2020.9340842.

[16] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

[17] Han Sun, Yizhao Wang, Zhenning Zhou, Shuai Wang, Haibo Yang, Jingyuan Sun, and Qixin Cao. Exploring pose-guided imitation learning for robotic precise insertion, 2025. URL https://arxiv.org/abs/2505.09424.

[18] Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. Chatgpt for robotics: Design principles and model abilities, 2023. URL https://arxiv.org/abs/2306.17582.

[19] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16606–16616, 2021.

doi: 10.1109/CVPR46437.2021.01634.

[20] 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*, pages 17868–17879, 2024.

[21] Rosa Wolf, Yitian Shi, Sheng Liu, and Rania Rayyes. Diffusion models for robotic manipulation: A survey, 2025. URL https://arxiv.org/abs/2504.08438.

[22] Jun Yang, Wenjie Xue, Sahar Ghavidel, and Steven L. Waslander. Active 6d pose estimation for textureless objects using multi-view rgb frames, 2025. URL https://arxiv.org/abs/2503.03726.

[23] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1941–1950, 2019.