ActivePose: Geometry-Grounded Confidence-Aware Active Perception for Robotic Manipulation

Sheng Liu
Karlsruhe Institute of Technology, Germany

***Abstract*—Robust robot manipulation requires more than estimating a task state: a robot must also recognize when its observation is insufficient and take action to obtain a more informative one. We present *ActivePose*, a geometry-grounded, confidence-aware active perception framework for 6-DoF object pose estimation and tracking. For pose initialization, a vision–language model (VLM) is grounded with entropy-ranked CAD renderings that encode which viewpoints are geometrically ambiguous. At test time, the system assesses whether the current observation supports a reliable pose estimate; when confidence is low, it imagines IK-feasible candidate views, fuses VLM ambiguity with pose-hypothesis entropy, and executes a next-best-view action. During manipulation, a demonstration-trained diffusion policy generates receding-horizon camera trajectories that preserve visibility and recover from motion and occlusion. Experiments in simulation and on real dual-arm hardware show that this combination of semantic ambiguity assessment, geometric uncertainty, and active sensing substantially improves pose reliability and downstream peg-in-hole assembly. These results show that confidence-aware active perception can convert pose uncertainty into actionable sensing decisions for CAD-available manipulation tasks.**

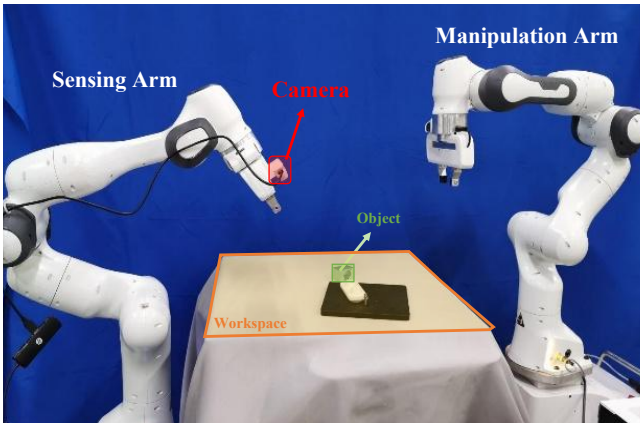

Fig. 1. Dual-arm setup. A sensing arm carries a wrist-mounted RGB-D camera, while a manipulation arm interacts with the object. Active perception lets the sensing arm change viewpoint when the current observation is ambiguous or likely to lose visibility.

I. Introduction

Learned robotic systems are commonly evaluated by whether they can infer a task-relevant state and produce an action. In real deployments, however, an equally important capability is to determine whether the available observation is *sufficient* for that inference. A confident but incorrect 6D pose estimate can cause grasping or assembly to fail, while a fixed sensor may lose the target as manipulation changes the scene. These failures are especially common for symmetric, textureless industrial parts and under self-occlusion or inter-object occlusion.

This paper studies active perception as a form of confidence-aware robot behavior. Rather than treating perception as a passive front end, the robot estimates whether its current evidence admits a unique pose and, when it does not, changes its viewpoint to acquire more informative evidence. The central challenge is that geometric ambiguity is object- and viewpoint-dependent: generic visual reasoning alone is not reliably calibrated to the symmetries encoded by a particular CAD model, while a single numerical uncertainty score can be brittle under rendering-to-real gaps and partial occlusions.

We propose *ActivePose*, a closed-loop framework that combines semantic ambiguity assessment with geometry-driven uncertainty. A VLM is grounded using CAD renderings selected from the lowest- and highest-entropy pose hypotheses, providing object-specific examples of unambiguous and ambiguous observations. At test time, the VLM assesses the current view; if ambiguity is detected, the robot renders IK-feasible candidate observations, fuses VLM ambiguity with FoundationPose hypothesis entropy, and executes the lowest-uncertainty next-best view. Once initialized, a demonstration-trained diffusion policy actively moves a wrist camera to preserve visibility under object motion and temporary occlusion.

Our contributions are:

- A CAD-grounded ambiguity assessment module that uses entropy-ranked renderings to construct object-specific prompts, enabling a VLM to perform constrained binary assessment of whether the current observation supports a unique 6D pose.
- A confidence-aware active perception loop that converts ambiguity and pose uncertainty into an executable sensing decision by selecting an IK-feasible next-best view from robot-reachable candidate viewpoints.
- An uncertainty-fused view scoring strategy that combines VLM-based ambiguity cues with pose-hypothesis entropy, showing that object-specific visual ambiguity and geometric uncertainty provide complementary signals for active perception.
- A real-robot evaluation on dual-arm pose estimation, active tracking, and downstream peg-in-hole assembly, demonstrating improved robustness under pose ambiguity, motion, and occlusion compared with passive or non-adaptive sensing baselines.

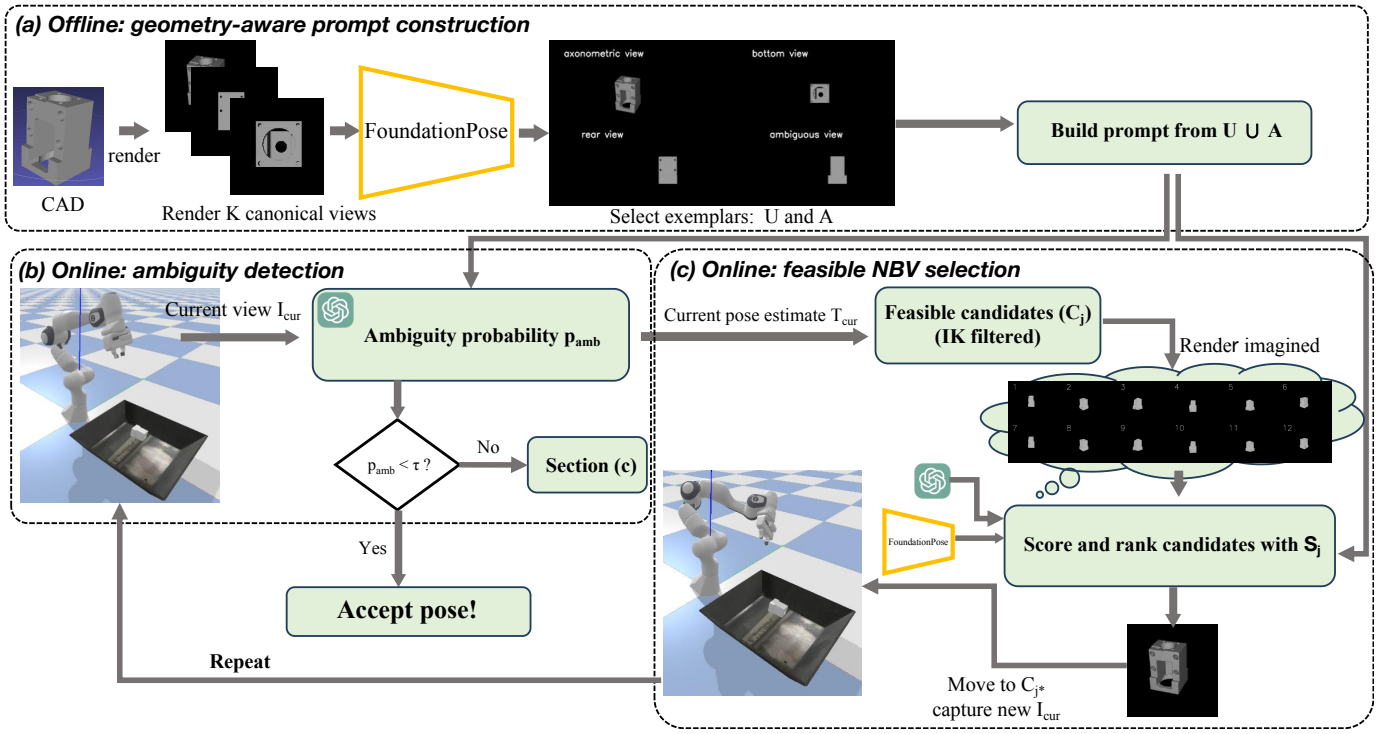

Fig. 2. **Active pose estimation.** (a) Offline: render canonical CAD views, compute the hypothesis entropy of FoundationPose, and build a geometry-aware prompt from low-/high-entropy exemplars. (b) Online: compute VLM ambiguity p_{amb} for the current view and trigger disambiguation when $p_{\mathrm{amb}} > \tau$. (c) Rank feasible candidate views using rendered imagined observations and the fused score, execute the selected NBV, and repeat up to budget L.

II. Related Work

A. Ambiguity and Active 6D Pose Estimation

Single-view 6D pose estimators have progressed from object-specific supervised methods [14, 25, 7, 13, 4, 15] to zero-shot systems that use CAD models for novel objects [11, 22, 9]. Yet a visually plausible observation can remain fundamentally ambiguous because of symmetry, textureless-ness, or occlusion. Active pose estimation addresses this by changing viewpoint, but existing methods often require object-specific training, hand-designed heuristics, or substantial annotations [16, 24, 10]. We instead use geometry-derived examples to teach a general VLM what ambiguity means for a specific object, then translate that assessment into closed-loop viewpoint selection.

B. Vision-Language Models for Constrained Perceptual Assessment

VLMs can connect visual observations with semantic concepts and have been used as task planners [3, 8, 19] or components of visuomotor policies [6, 21, 5]. Here the semantic question is not *what action achieves a language goal*, but *whether the current visual evidence is adequate for a geometric decision*. This is related to confidence-aware active perception: the robot estimates whether the current visual evidence is sufficient for reliable pose-conditioned action and invokes an information-gathering behavior when confidence is low. Geometry-ranked exemplars constrain the VLM to an object-specific binary ambiguity assessment, while pose-hypothesis entropy provides a complementary structured uncertainty signal.

C. Learned Active Sensing

Diffusion policies model multimodal action distributions and generate smooth receding-horizon behavior [23, 2]. Unlike work that uses diffusion models primarily to generate manipulation actions, we use one as an active sensing policy. It predicts camera trajectories that maintain target visibility and recover after occlusion, coupling learned behavior with an explicit perceptual objective.

We formulate active 6D pose estimation and tracking as a closed-loop decision problem. Given a CAD model, an initial camera pose, and an RGB-D stream, the robot must estimate both a task state—the object pose—and an evidence state— whether the current observation is sufficiently informative. During initialization, low confidence triggers a next-best-view action selected from imagined, kinematically feasible observations. Rather than solving continuous belief-space planning, we formulate view selection as a finite search over robot-executable candidate viewpoints generated around the current object estimate and filtered by kinematic feasibility. During manipulation, the sensing policy continually changes the camera pose to preserve visibility and avoid pose loss. This separation makes the confidence signal actionable: uncertainty is not only reported, but used to decide when and how the robot should sense again.

III. Method

Our approach contains two coupled modules. The first performs geometry-grounded self-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–assess–intervene 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 assessment with feasible NBV selection. The method assumes an object CAD model is available, which matches CAD-available manipulation settings such as industrial assembly.

Offline: We pre-render K canonical views $\{R_k\}_{k=1}^{K}$ by sampling camera poses on a tabletop-facing hemisphere around the object, with a fixed standoff distance d and the camera optical axis pointing to the object center. These renders are reused to construct an object-specific geometry-aware prompt that remains fixed for all online queries of the same object. The 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 [22] to obtain the top-N *6D* pose hypotheses $\{\widehat{T}_i\}_{i=1}^{N}$ with associated confidence weights $\{w_i\}_{i=1}^{N}$. 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 entropy 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. This construction grounds the VLM in an object-specific notion of pose ambiguity, rather than using it as an open-ended language planner or general semantic reasoning module.

Online: Given the current observation I_{cur}, we estimate an initial pose $T_{\text{cur}} = \text{FoundationPose}(I_{\text{cur}})$ and compute an ambiguity score p_{amb} using the fixed per-object prompt constructed offline. The VLM is queried with a constrained $\{\texttt{Yes}, \texttt{No}\}$ output format, where $\texttt{Yes}$ indicates that the current observation is ambiguous with respect to the object pose. We convert the corresponding log-probabilities into a normalized score:

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

We use this quantity as an operational ambiguity score for triggering active sensing, rather than as a formally calibrated probability across objects or domains.

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**
  $p_{\text{amb}} \leftarrow$ AmbiguityScore$(I_{\text{cur}}, \text{prompt})$ (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**
    $\hat{I}_j \leftarrow \text{Render}(C_j)$
    $p_{\text{amb},j} \leftarrow$ AmbiguityScore$(\hat{I}_j, \text{prompt})$ (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
${}^{c}T_o \leftarrow T_{\text{cur}}$

If $p_{\text{amb}} < \tau$, the observation is treated as sufficiently informative and we directly output T_{cur}; otherwise, we iteratively select a feasible NBV within a small budget L. A larger τ makes disambiguation less frequent, since only highly ambiguous views trigger additional sensing, while a smaller τ triggers NBV more aggressively.

Given the current camera pose ${}^{b}T_c$ and the current pose estimate T_{cur}, we first convert the estimated object pose into the robot base frame as ${}^{b}T_o = {}^{b}T_c \cdot T_{\text{cur}}$. The generator $\mathcal{G}$ then samples M *local* candidate camera poses around the estimated object pose and retains candidates that are executable by the sensing arm. In our implementation, feasibility is defined by the availability of an IK solution satisfying joint limits and predefined workspace constraints. Thus, view selection is formulated as a finite search over robot-executable candidate viewpoints rather than continuous belief-space planning.

For each feasible candidate C_j, we render an imagined view $\hat{I}_j$ and compute the VLM-based ambiguity score $p_{\text{amb},j}$ with

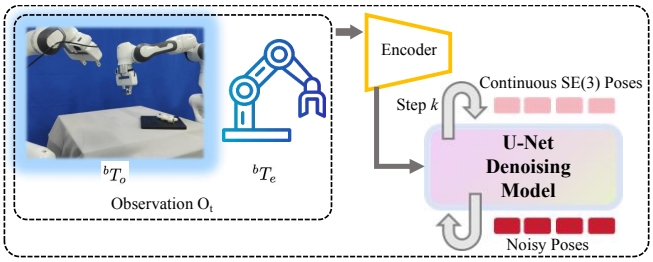

Fig. 3. **Active pose tracking.** The policy encodes the current observation O_t, denoises over K_d reverse-diffusion steps to generate a horizon of continuous SE(3) poses, and executes the last k_h poses in a receding-horizon loop.

the same object-specific prompt. We also compute the pose-hypothesis entropy $\bar{H}(\hat{I}_j)$ from FoundationPose hypotheses. Candidate views are scored by

$$S_j = \lambda\, \bar{H}(\hat{I}_j) + (1-\lambda)\, p_{\mathrm{amb},j}, \tag{3}$$

where the first term captures structured geometric uncertainty and the second term captures CAD-grounded visual ambiguity. The robot moves the camera to the lowest-scoring candidate view, acquires a new real observation, and updates the pose estimate. We repeat this process until the newly acquired image yields $p_{\mathrm{amb}} < \tau$ or the budget L is exhausted (Algorithm 1); at termination, we return the current estimate as the final pose ^cT_o.

B. Active Pose Tracking

Algorithm 2 Active Pose Tracking

Require: Object pose in camera frame ^cT_o, initial end-effector pose $^bT_{e,1}$, camera pose in end-effector frame ^eT_c, history length H, prediction horizon K, executed tail length k_h, total steps N.

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

$t \leftarrow H$
while $t \leq N$ **do**
  Compute object pose in base frame: ^bT_o (Eq. (4))
  Collect history window: O_t
  Sample a future end-effector pose sequence by denoising:
  $$\{^b\widehat{T}_e^{\,t+1:t+K}\} \leftarrow \mathrm{DP}(O_t)$$
  ▷ DP denotes the diffusion policy, see Eqs. (5)–(6)
  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 ^bT_e to $\mathcal{T}_t$, then execute the resulting continuous trajectory via servo control.
  Update ^bT_e to the last executed pose: $^bT_e \leftarrow {}^b\widehat{T}_e^{\,t+K}$
  $t \leftarrow t + k_h$
end while

After obtaining the final disambiguated pose ^cT_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 ^eT_c, a predicted end-effector trajectory $\{^bT_{e,t}\}$ uniquely induces a camera trajectory $\{^bT_{c,t}\}$ via $^bT_c = {}^bT_e \cdot {}^eT_c$. 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 ^cT_o. We convert it to the robot base frame using the current end-effector pose ^bT_e and the fixed camera extrinsic ^eT_c:

$$^bT_o = {}^bT_c \cdot {}^cT_o = {}^bT_e \cdot {}^eT_c \cdot {}^cT_o. \tag{4}$$

We then build the observation $O_t = \{^bT_o^{\,t-H+1:t},\ ^bT_e^{\,t-H+1:t}\}$, containing the past H frames of (i) the object pose in the robot base frame ^bT_o and (ii) the end-effector pose in the base frame ^bT_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:

$$A_t^{k-1} = \alpha_k\Big(A_t^k - \gamma_k\, \varepsilon_\theta(O_t, A_t^k, k)\Big) + \sigma_k\, \mathcal{N}(0, I), \tag{5}$$

$$\mathcal{L} = \mathbb{E}\Big[\big\|\epsilon^k - \varepsilon_\theta\big(O_t,\ \hat{\alpha}_k A_t^0 + \hat{\beta}_k \epsilon^k,\ k\big)\big\|^2\Big], \tag{6}$$

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 [17]. Mean squared error (MSE) loss is employed as the objective function to supervise noise prediction (Eq. (6)). The rotation is represented as R6D [20], 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 ^bT_e to this segment for smooth, feasible motion and repeat in a receding-horizon loop.

IV. 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 [12], 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:

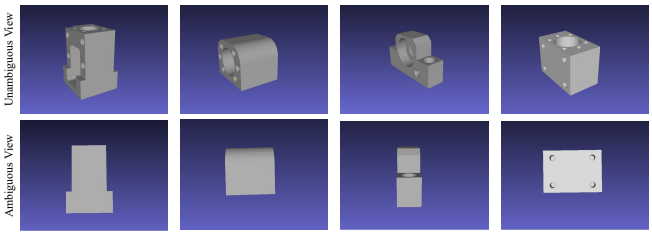

Fig. 4. CAD models of the experimental objects (Obj. 1–3 from MP6D; Obj. 4 for assembly) and an example ambiguous view.

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. All experiments are conducted in CAD-available manipulation settings, where the object geometry is known in advance and can be used for prompt construction and rendering. This setting matches industrial assembly scenarios in which CAD models are typically available, but it does not address open-vocabulary ambiguity reasoning over arbitrary unseen objects.

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. Candidate viewpoints are sampled locally around the current object estimate and filtered by sensing-arm kinematic feasibility before NBV scoring. Thus, our view selection procedure is a finite search over robot-executable candidate viewpoints, rather than a continuous active planning formulation that jointly optimizes sensing cost, collision avoidance, and downstream manipulation objectives. 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 use this setting to evaluate whether the proposed perception loop can benefit from task-specific active sensing behaviors learned from limited supervision. We do not claim broad cross-object or cross-task policy generalization from these demonstrations. 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.

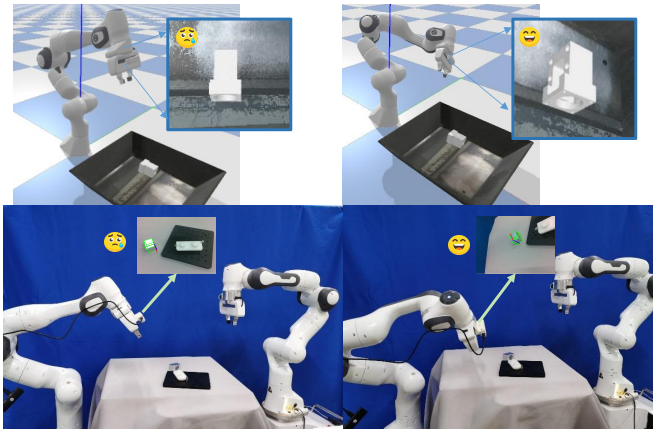

Fig. 5. **Example of active pose estimation.** An ambiguous initial view triggers NBV selection; moving to the selected viewpoint yields an unambiguous 6D pose estimate, shown in simulation (top) and real-robot trials (bottom).

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).

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.

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 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

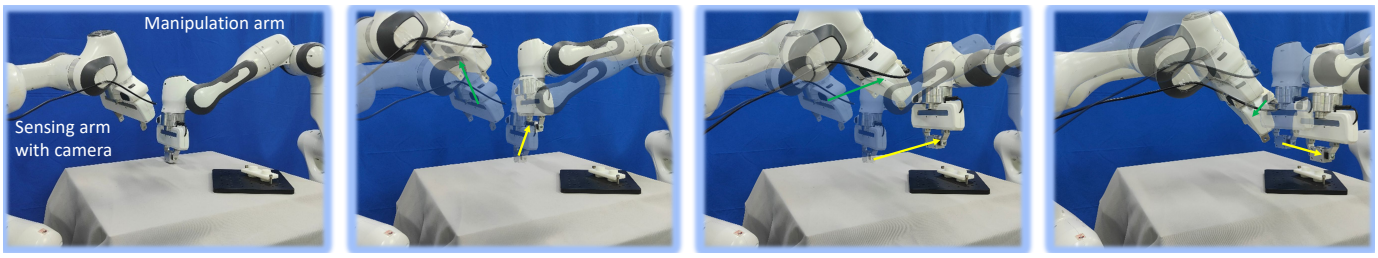

Fig. 6. **Example of active pose tracking.** While the manipulation arm manipulates the object, the sensing arm with a wrist-mounted RGB-D camera actively follows to keep the target in view.

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

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 $\pm$ 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%

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 [18], 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.

V. Discussion: Confidence-Aware Active Perception

ActivePose uses a VLM for a deliberately narrow perceptual judgment: whether the current observation is ambiguous for the object-specific 6D pose. The entropy-ranked CAD renders act as grounded examples of ambiguous and unambiguous viewpoints, rather than generic visual prompts or open-ended language instructions. This design constrains the VLM to assess perceptual sufficiency in a CAD-available setting, where the relevant ambiguity arises from geometry, symmetry, texturelessness, and occlusion.

The ablations support this interpretation. Text-only prompting or randomly selected CAD examples reduces performance, indicating that the VLM benefits from object-specific geometric grounding. At the same time, VLM-based ambiguity cues alone are weaker than their fusion with pose-hypothesis entropy, while entropy alone can fail to identify views that remain visually misleading for manipulation. These results suggest that CAD-grounded visual ambiguity and structured geometric uncertainty provide complementary signals: the former captures whether a view appears pose-ambiguous, while the latter summarizes the spread of candidate pose hypotheses.

More broadly, the framework illustrates a practical confidence-aware active perception pattern for robotics: estimate the task-relevant state, assess whether the available evidence is sufficient for reliable action, and gather additional information when confidence is low. This differs from treating uncertainty as a post-hoc diagnostic quantity. In ActivePose, the confidence signal directly changes the robot's sensing behavior by triggering next-best-view actions during initialization and visibility-preserving camera motions during manipulation.

VI. Limitations

ActivePose assumes that a CAD model is available and constructs object-specific prompt examples from rendered views; it therefore targets CAD-available manipulation settings such as industrial assembly, rather than open-vocabulary ambiguity reasoning over arbitrary objects. Extending the method to objects without CAD models would require alternative sources of object-specific geometry or learned priors for ambiguity.

The VLM log-probabilities are used as operational ambiguity scores, not as calibrated or formally guaranteed uncertainties. Their values may vary across objects, viewpoints, and the render-to-real gap between CAD images and real camera observations. In this work, these scores are used for ranking and triggering active sensing within a fixed object-specific prompt, but future work should study calibration, domain adaptation, or distilled object-specific ambiguity classifiers for more predictable deployment.

Candidate viewpoints are sampled from a finite local set and filtered for inverse-kinematics feasibility. This makes the system executable on the real robot, but the current formulation does not jointly optimize collision risk, sensing cost, arm coordination, or downstream task value. Incorporating continuous, collision-aware, task-coupled view planning would make the method more general, especially in cluttered workspaces or tightly constrained dual-arm manipulation settings.

The active tracking policy is trained from ten demonstrations per task and has not been evaluated for broad cross-object or cross-task transfer. The tracking experiments therefore demonstrate task-specific active sensing learned from limited supervision, rather than general-purpose policy transfer. Finally, experiments cover four objects and a single peg-in-hole assembly case study; larger-scale testing across more object categories, workspace layouts, and manipulation tasks is needed to characterize rare failures and robustness under production variability.

VII. Conclusion

We presented ActivePose, a confidence-aware active perception system for CAD-available robotic manipulation. Entropy-ranked CAD examples ground a VLM for constrained ambiguity assessment, pose-hypothesis entropy provides a complementary geometric uncertainty signal, and their fusion triggers feasible next-best-view actions when the current observation is insufficient. A diffusion policy further maintains visibility during manipulation.

Real-robot experiments show that this closed loop improves pose estimation, active tracking, and downstream peg-in-hole assembly compared with passive or non-adaptive sensing baselines. These results suggest that active perception for manipulation should not only estimate task-relevant states, but also assess whether the current evidence is sufficient for reliable action.

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] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators, 2024. URL https://arxiv.org/abs/2311.01378.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] Han Sun, Sheng Liu, Yizhao Wang, Zhenning Zhou, Shuai Wang, Haibo Yang, Jingyuan Sun, and Qixin Cao. Exploring pose-guided imitation learning for robotic precise insertion, 2026. URL https://arxiv.org/abs/2505.09424.

[19] 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.

[20] 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.

[21] Sicheng Wang, Sheng Liu, Weiheng Wang, Jianhua Shan, and Bin Fang. Robobert: An end-to-end multimodal robotic manipulation model, 2025. URL https://arxiv.org/abs/2502.07837.

[22] 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.

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

[24] 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.

[25] 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.