RECOVERING THE VIEW: BENCHMARKING PHYSICAL ACTIVE VISION FOR OCCLUSION RECOVERY IN ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

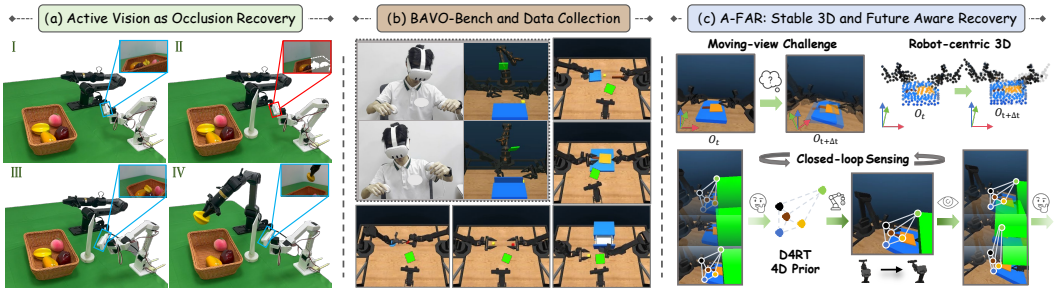

Figure 1: **Overview of our motivation, benchmark, and approach. (a)** We study physical active vision as closed-loop recovery from external occlusion: using a single active camera as the sole visual sensor, the robot actively changes its viewpoint to recover task-relevant visibility and continue manipulation. **(b)** We introduce **BAVO-Bench**, a bimanual active-vision manipulation benchmark that systematically introduces controlled external occlusions to study viewpoint recovery, together with a VR teleoperation pipeline for collecting active-view demonstrations. **(c)** We present **A-FAR**, an active-vision policy that canonicalizes moving-view observations in robot-centric 3D and distills D4RT's 4D relational priors, enabling stable, future-aware control under active viewpoint changes.

ABSTRACT

Physical active vision allows robots to change their viewpoint when task-relevant observations become unreliable, yet existing manipulation benchmarks provide limited support for studying how policies recover from occlusion during execution. We introduce **BAVO-Bench (Bimanual Active Vision under Occlusion)**, a bimanual active-vision benchmark that systematically controls external visibility through Clean, Stage Occlusion, and Random-time Occlusion conditions, enabling evaluation of both manipulation performance and active visual recovery. Building on this setting, we present **A-FAR (Active Future-Aware Recovery)**, an active-vision policy for joint viewpoint and manipulation control. A-FAR represents moving-camera observations in a unified robot-centric 3D frame and distills relational structure together with its future evolution from a pretrained 4D model, providing the policy with future-aware geometric guidance without requiring future observations at deployment. Experiments across multiple manipulation tasks show that A-FAR improves robustness to both structured and temporally shifted occlusions while maintaining strong performance under clean observations.

1 INTRODUCTION

Robotic manipulation depends not only on how to act, but also on what the robot can see. Modern robot policies now range from task-specific visuomotor policies to generalist Vision-Language-Action (VLA) models and emerging World Action Models (WAMs), yet most still rely on passive visual observations from fixed or wrist-mounted cameras (Zhao et al., 2023; Chi et al., 2023; Kim et al., 2025; Black et al., 2025b;a; Ye et al., 2026). Such configurations can become unreliable when

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.

task-relevant regions are occluded or fall outside the available viewpoints, while wrist-mounted cameras further couple perception with manipulation motion. Recent methods have explored computational alternatives that select or synthesize virtual observations from existing scene evidence (Liu et al., 2026b; Heo et al., 2026). However, their effectiveness depends on the fidelity of the reconstructed or generated scene, particularly when task-relevant regions are heavily occluded or poorly observed. Physical active vision instead moves the sensor to a new line of sight, allowing the robot to acquire fresh measurements and actively restore task-relevant visibility (Chuang et al., 2025; Xiong et al., 2025).

Despite this progress, existing active-vision manipulation largely studies viewpoint requirements induced by task geometry or execution stages. For example, the robot may need to change its viewpoint to reveal an occluded target or acquire task-relevant visual evidence as execution progresses (Wang et al., 2025; Li et al., 2026). Prior work has further extended active-perception evaluation to occlusion-aware manipulation settings (He et al., 2026; Liu et al., 2026a; Cheng et al., 2018). Yet task-relevant visibility can also be disrupted unexpectedly during execution, for example when a human hand or a moved object blocks the current view. The policy must then recover an informative viewpoint before continuing manipulation. We therefore study physical active vision as closed-loop perceptual recovery from externally induced visibility failures during execution (Fig. 1(a)).

To systematically evaluate this capability, we introduce **BAVO-Bench** (Bimanual Active Vision under Occlusion), a benchmark for bimanual manipulation with physical active vision built on the AV-ALOHA simulation platform (Chuang et al., 2025). BAVO-Bench contains five multi-stage bimanual tasks and treats external occlusion as an explicit, controllable intervention while preserving the underlying task and control interface. Each task is evaluated under three visibility conditions: **Clean**, without artificial occlusion; **Stage Occlusion**, where task-relevant occlusions are introduced at structured stages and used for training demonstrations; and **Random-time Occlusion**, where similar visibility failures occur at temporally unexpected moments during execution. This design separates nominal manipulation, recovery under the demonstration distribution, and generalization to unexpected occlusion timing. Since task success alone cannot determine whether the policy actually recovered a useful view, we evaluate task progress and introduce **Camera-attributable Visibility Gain (CVG)** to measure whether camera motion itself restores task-relevant visibility (Fig. 1(b)).

Physical active vision also changes the learning problem itself. Camera motion can substantially change the observation even when the underlying scene remains unchanged. We therefore represent RGB-D observations as colored point clouds in a shared robot-base frame and build our policy on ManiFlow (Yan et al., 2025) to obtain a more stable representation under viewpoint changes. A more fundamental challenge is that the policy also controls the sensor that generates its future observations. A useful viewpoint action must therefore support not only the current observation but also subsequent manipulation, which requires anticipating how task-relevant spatial structure may evolve. To address these challenges, we introduce **A-FAR (Active Future-Aware Recovery)**, an active-vision policy that combines robot-centric 3D observations with temporal relational supervision from D4RT (Zhang et al., 2026a). During training, D4RT supervises current point relations and their future evolution, allowing A-FAR to learn future-aware scene representations for joint viewpoint and manipulation control without requiring the 4D teacher at deployment (Fig. 1(c)).

In summary, our contributions are threefold:

- We identify recovery from externally induced visibility failures as a distinct setting for physical active vision and introduce **BAVO-Bench**, a five-task bimanual active-vision benchmark with controlled occlusion timing, together with **Camera-attributable Visibility Gain (CVG)** for measuring visibility recovery due to camera motion.
- We propose **A-FAR**, a viewpoint-stable and future-aware active-vision policy that combines robot-centric 3D observations with training-time 4D relational distillation for joint viewpoint and manipulation control.
- We develop a low-cost, VR-enabled active-vision stack for unified demonstration collection in simulation and the real world, with a physical active-camera platform for deployment.

2 Related Work

Physical active vision for robotic manipulation. Active perception enables robots to actively change what they observe during manipulation. Early work jointly learned camera and manipulation actions under occlusion (Cheng et al., 2018), while recent systems learn coordinated viewpoint control through movable sensors, including AV-ALOHA (Chuang et al., 2025), Vision in Action (Xiong et al., 2025), and SaPaVe (Liu et al., 2026a). Most recently, ActiveScale scales this capability across model, data, and hardware through pose-grounded temporal modeling, large-scale egocentric training, and an independently actuated camera platform (Zhou et al., 2026). Computational approaches such as ActiveVLA instead select informative viewpoints from reconstructed scene representations (Liu et al., 2026b). Our work focuses on physical active vision as *closed-loop recovery*: restoring task-relevant visibility after it is externally disrupted during ongoing manipulation.

Benchmarks for active-view manipulation. Recent benchmarks increasingly make information acquisition an explicit manipulation capability. EFM-10 organizes tasks around exploratory and focused perception (He et al., 2026), ActiveManip-Bench evaluates unoccluded, occluded, and out-of-view conditions (Liu et al., 2026a), and TAVIS studies active sensing under procedural shifts and evaluates anticipatory gaze (Spigler, 2026). In contrast, we treat visibility failure as a controllable *within-episode intervention* over an otherwise unchanged task. Stage Occlusion evaluates demonstrated recovery structure, while Random-time Occlusion breaks the correlation between task progress and intervention onset. Our CVG metric further compares the recovered view with a same-state frozen-camera counterfactual, attributing visibility improvement specifically to camera motion.

Viewpoint-consistent and future-aware representations. Active camera motion creates both cross-view observation variation and coupling between viewpoint actions and future observations. ActiveScale addresses cross-view reasoning using temporal images and explicit camera-pose supervision (Zhou et al., 2026), while EgoAVFlow jointly predicts future 3D flow and camera trajectories for active vision (Cho et al., 2026). Closely related to our relational supervision, MECo-WAM distills current and temporal 4D geometric relations into an inference-efficient policy (Zhang et al., 2026b). A-FAR instead targets active-view recovery: we express RGB-D observations in a shared robot-centric frame and use point-aligned current/future queries from D4RT (Zhang et al., 2026a) to supervise a student-side predictor of relational evolution, which conditions joint viewpoint–manipulation control without requiring future observations or the 4D teacher at deployment.

3 BAVO-Bench: Physical Active-View Recovery under External Occlusion

We build **BAVO-Bench** on the AV-ALOHA active-vision simulator (Chuang et al., 2025). The original AV-ALOHA benchmark primarily studies tasks whose viewpoint requirements arise from task geometry or predictable execution stages. To focus on recovery from externally induced visibility loss, we design five multi-stage tasks around common bimanual manipulation patterns and introduce external occlusion as a controlled intervention, as illustrated in Fig. 2.

3.1 Controlled Visibility Interventions

The three conditions differ in whether and when external occlusion is introduced. At each intervention, a task-relevant target is selected according to task progress, and a camera-facing occluder is placed along its line of sight from the active camera. Placements are validated for target occlusion and robot clearance, and the occluder remains fixed in the world frame until the next intervention. Task definitions and additional intervention details are provided in Appendix A.

Clean. Clean denotes nominal task execution without benchmark occlusion. It serves as the reference for the policy's manipulation performance under standard visual conditions.

Stage Occlusion. Each episode begins with an initial task-relevant occlusion. As execution progresses, the occlusion target is updated at two predefined task transitions, producing three structured occlusion phases over the trajectory. Because these transitions are fixed across demonstrations, Stage provides the training-matched setting for learning and evaluating active-view recovery.

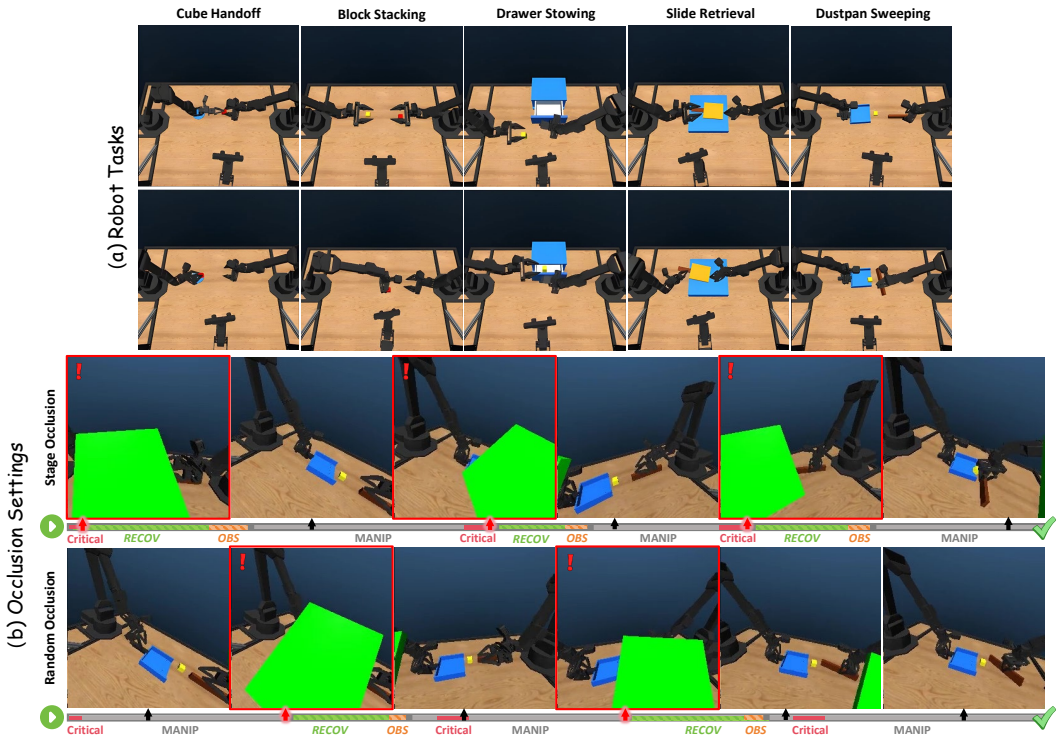

Figure 2: **BAVO-Bench overview. Top:** Five multi-stage bimanual manipulation tasks. **Bottom:** Representative rollouts under the two occlusion settings. Red-bordered frames mark occlusion events, and the timelines illustrate recovery and manipulation behaviors. Both settings occlude task-relevant targets; **Stage Occlusion** aligns interventions with predefined task transitions, whereas **Random-time Occlusion** samples intervention times independently of these transitions.

Random-time Occlusion. Each episode likewise begins with an initial occlusion, but subsequent interventions are triggered at sampled times that need not coincide with the predefined transitions used in Stage. This breaks the regular association between task progression and viewpoint recovery, testing whether the policy responds to visibility failures themselves rather than replaying stage-specific camera behavior. Random-time may additionally contain more interventions than Stage and is therefore treated as a stronger recovery condition rather than a strictly timing-matched ablation.

3.2 Demonstration and Evaluation Protocol

We collect 100 VR demonstrations per task under Stage Occlusion, yielding 500 trajectories in total. Following a new occlusion, operators are encouraged to primarily reposition the observation arm while keeping manipulation approximately stable; once an informative viewpoint is recovered, the observation arm remains comparatively stable and task execution continues. This provides structured demonstrations of physical view recovery followed by manipulation.

Policies are trained exclusively on Stage demonstrations and evaluated using the same trained model under all three conditions. This protocol evaluates nominal manipulation under Clean, training-matched recovery under Stage, and recovery under shifted intervention timing in Random-time.

3.3 Evaluation Metrics

We report Success Rate (SR) and Task Progress Score (TPS) for task performance, together with Camera-attributable Visibility Gain (CVG) to directly evaluate visibility recovery attributable to viewpoint change.

Success Rate and Task Progress. Success Rate (SR) measures the fraction of episodes that complete the full task. To capture partial completion, we additionally report Task Progress Score (TPS).

Each task contains five ordered milestones; if $d_i \in \{0, \ldots, 5\}$ denotes the deepest milestone reached in episode i, then $\mathrm{TPS} = \frac{1}{N}\sum_{i=1}^{N} d_i/5$. Together, SR and TPS characterize task performance at complementary granularities: final task completion and intermediate progress.

Camera-attributable Visibility Gain. To isolate the sensing consequence of viewpoint change, we introduce Camera-attributable Visibility Gain (CVG). Consider an occlusion event e introduced at time τ_e, and let x_t denote the current robot and object configuration. For an evaluation camera pose C, we measure normalized target visibility as

$$V_e(C, x_t) = \frac{N_e^{\mathrm{occ}}(C, x_t)}{N_e^{\mathrm{noOcc}}(C, x_t)}, \tag{1}$$

where N_e^{occ} and N_e^{noOcc} are the visible target pixels rendered from the same camera and scene state with and without the benchmark occluder, respectively. The ratio therefore measures how much of the target that would otherwise be visible remains visible under the external occlusion.

CVG compares the policy's realized camera trajectory with a counterfactual reference in which the camera remains fixed at its event-start pose C_{τ_e}. Over a recovery window of H steps,

$$\mathrm{CVG}_e = \frac{1}{H}\sum_{h=0}^{H-1}\left[V_e(C_{\tau_e+h}, x_{\tau_e+h}) - V_e(C_{\tau_e}, x_{\tau_e+h})\right]. \tag{2}$$

Crucially, both visibility terms are evaluated on the same current scene state x_{τ_e+h}. Changes caused by manipulation or object motion are therefore shared by both views, while their difference captures the visibility advantage associated with viewpoint change. A positive CVG indicates that the policy-selected camera trajectory reveals more of the task-relevant target than would have been visible had the camera remained at its event-start pose.

The counterfactual renders are used only for privileged simulation-time evaluation and never affect policy observations or actions. Recovery-window and aggregation details for CVG evaluation are provided in Appendix A.

4 A-FAR: Active Future-Aware Recovery

We present **A-FAR**, a ManiFlow-based policy (Yan et al., 2025) for joint manipulation and active-view control. As illustrated in Fig. 3, A-FAR combines a viewpoint-stable 3D observation with training-time 4D relational supervision. A lightweight predictive branch distills this supervision into future-aware point representations and injects them into the action policy.

4.1 Viewpoint-Stable 3D Observation

Active-view control benefits from a visual representation that remains geometrically consistent as the sensor moves. In image space, camera motion can induce large changes in pixel location and appearance even for static scene geometry, leaving the policy to disentangle viewpoint-induced variation from task-relevant changes. Lifting RGB-D observations into 3D removes this image-plane dependence, but a camera-frame point cloud still moves with the sensing coordinate system. We therefore express all observed geometry in a shared robot-base frame.

Specifically, let $T_{B\leftarrow E_t} \in SE(3)$ denote the rigid transformation from the observation-arm end-effector frame E_t to the robot-base frame B, obtained from forward kinematics at configuration q_t^{view}. Let $T_{E\leftarrow C} \in SE(3)$ denote the fixed camera-to-end-effector transformation obtained from hand–eye calibration. Their composition gives the camera-to-base transformation $T_{B\leftarrow C_t} = T_{B\leftarrow E_t}T_{E\leftarrow C}$. We back-project the current RGB-D observation and transform the resulting points by $T_{B\leftarrow C_t}$, obtaining a colored point cloud $P_t \in \mathbb{R}^{N\times 6}$. In this shared frame, camera motion no longer induces a global displacement of static geometry, while object motion, occlusion, disocclusion, and newly revealed surfaces remain explicit.

4.2 Training-only 4D Relational Teacher

Physical active vision couples action generation with future perception: viewpoint actions determine where the camera moves, and therefore alter the observations available to subsequent manipulation.

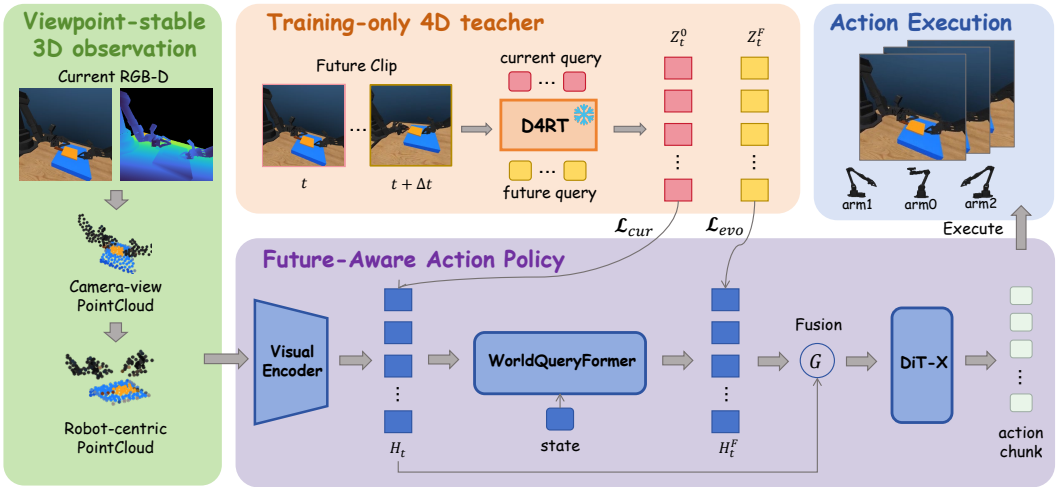

Figure 3: **Overview of A-FAR.** Current RGB-D observations are canonicalized into a robot-base point cloud and encoded as current point tokens H_t. A state-conditioned WorldQueryFormer predicts future relational tokens $\hat{H}_t^F$, which are fused with the current representation for ManiFlow action generation. During training, a frozen D4RT teacher queries the same source points at current and future target times to construct relational targets that supervise the current structure ($\mathcal{L}_{\text{cur}}$) and its temporal evolution ($\mathcal{L}_{\text{evo}}$). The teacher branch and future frames are removed at deployment.

A policy that only represents the current scene must learn this coupling implicitly, without an signal of how task-relevant spatial structure evolves along the trajectory. We instead exploit the demonstrated future as privileged supervision. Specifically, we introduce Dynamics-Aware Temporal Relational Distillation, using a frozen OpenD4RT teacher (Li & RHOS Team, 2026), an unofficial implementation of D4RT (Zhang et al., 2026a), instantiated with OpenD4RT, to extract point-aligned supervision over current scene relations and their future evolution.

Point-aligned queries. Each queried point retains its source pixel (u_i, v_i) in the current frame t. Using D4RT's query format $q = (u, v, t_s, t_t, t_c)$, we query the same source point under current and future temporal contexts:

$$q_i^0 = (u_i, v_i, t, t, t), \qquad q_i^F = (u_i, v_i, t, t + \Delta t, t + \Delta t). \tag{3}$$

The corresponding decoder features $Z_t^0, Z_t^F \in \mathbb{R}^{N_q \times d_T}$ form point-aligned current–future pairs, where N_q is the number of queried anchor points and d_T is the teacher feature dimension. This construction follows the point-query interface used by D4RT to probe scene information across space and time.

Relational targets. Directly matching teacher features would require the student to reproduce the teacher's latent coordinates, whereas our goal is to transfer the relational geometry encoded in that feature space. We therefore distill pairwise distances among point-aligned teacher features. For feature sets $X = \{x_i\}$ and $Y = \{y_i\}$, we define

$$\mathcal{R}(Y \mid X)_{ij} = \frac{\|y_i - y_j\|_2}{\text{mean}_{p<q} \|x_p - x_q\|_2 + \epsilon}, \qquad R_t^T = \mathcal{R}(Z_t^0 \mid Z_t^0), \qquad \Delta R_t^T = \mathcal{R}(Z_t^F \mid Z_t^0) - R_t^T. \tag{4}$$

Here, R_t^T describes the pairwise geometry of the queried tokens in D4RT's current feature space, while ΔR_t^T captures how this feature-space geometry evolves along the demonstrated trajectory. This relational formulation transfers the teacher's latent structure without requiring channel-wise alignment between D4RT and the policy representation.

Most feature pairs exhibit only weak relational changes over time, which can dilute the supervision carried by interaction-relevant dynamics. We therefore weight the evolution objective according to the magnitude of teacher-predicted relation changes, emphasizing pairs with stronger temporal variation while retaining supervision over the remaining relational structure. The exact weighting formulation is deferred to Appendix B.

4.3 FUTURE-AWARE ACTION POLICY

The relational targets specify what temporal structure should be captured, but this information is useful for control only if the policy can predict and exploit it from the current observation. The ManiFlow visual encoder maps P_t to current point tokens $H_t \in \mathbb{R}^{N\times d}$, from which a lightweight WorldQueryFormer predicts point-aligned future tokens $\hat{H}_t^F = \mathcal{F}_\phi(H_t, s_t)$ conditioned on the robot state s_t. As shown in Fig. 3, WorldQueryFormer treats the current point representation as observation memory and refines state-conditioned point hypotheses through two lightweight blocks of inter-point self-attention and cross-attention to that memory. The predicted tokens are supervised by the relational evolution introduced in Sec. 4.2, while architectural details are deferred to Appendix B.

Rather than using the predicted representation only as an auxiliary training target, we inject it into the action pathway through a zero-initialized feature-wise gated residual:

$$H_t^{\text{cond}} = H_t + \tanh(\gamma) \odot W_F \hat{H}_t^F. \tag{5}$$

This initialization preserves the original current-observation pathway at the start of training, while allowing the policy to learn how strongly predictive features should influence control. The fused representation H_t^{cond}, together with the robot state, conditions ManiFlow's DiT-X to generate both manipulation and viewpoint actions.

4.4 TRAINING OBJECTIVE AND INFERENCE

We apply the same relational operator to the normalized current and predicted student tokens, yielding the student current relation R_t^S and evolution ΔR_t^S. A current-relation loss shapes the visual representation, while a dynamics-weighted evolution loss additionally trains the future-token predictor. Together with the ManiFlow action objective, the full loss is

$$\begin{aligned} \mathcal{L}_{\text{cur}} &= \|R_t^S - R_t^T\|_1, \\ \mathcal{L}_{\text{evo}} &= \|\Delta R_t^S - \Delta R_t^T\|_{1,\omega}, \\ \mathcal{L} &= \mathcal{L}_{\text{ManiFlow}} + \frac{\lambda_{\text{rel}}}{2}\left(\mathcal{L}_{\text{cur}} + \mathcal{L}_{\text{evo}}\right), \end{aligned} \tag{6}$$

where $\|\cdot\|_{1,\omega}$ denotes the dynamics-aware weighted L1 loss. The action objective is evaluated on the fused representation H_t^{cond}, allowing the distilled future structure to be shaped by its usefulness for manipulation and viewpoint control.

Teacher targets are precomputed from demonstration clips. At deployment, D4RT and future observations are removed; A-FAR requires only the current RGB-D observation and robot state.

5 EXPERIMENTS

We evaluate A-FAR on BAVO-Bench to examine both task performance and active-view behavior under externally induced visibility disruptions. We first compare A-FAR with representative visuomotor and vision-language-action baselines across Clean, Stage Occlusion, and Random-time Occlusion. We then analyze how visibility recovery relates to downstream task execution, followed by ablations on future-aware relational modeling and viewpoint stabilization. Finally, we demonstrate the feasibility of deploying the full policy on our real-world active-camera platform.

5.1 EXPERIMENTAL SETUP

Setup. We evaluate ACT (Zhao et al., 2023), ManiFlow (Yan et al., 2025), $\pi_{0.5}$ (Black et al., 2025a), and A-FAR on the five BAVO-Bench tasks under Clean, Stage Occlusion, and Random-time Occlusion. Each task contains 100 Stage-Occlusion demonstrations. ACT, ManiFlow, and A-FAR are trained per task with three random seeds and evaluated for 20 episodes per condition; due to its substantially higher training cost, $\pi_{0.5}$ is trained once on the mixed five-task dataset.

Metrics. We report success rate (SR) and task progress score (TPS) under all conditions, and Camera-attributable Visibility Gain (CVG) under Random-time Occlusion. SR and TPS are reported in percentages, while CVG is reported as $100 \times$ CVG in percentage points (pp). Unless otherwise specified, results are macro-averaged across the five tasks.

Table 1: **Main results on BAVO-Bench.** Results are macro-averaged over five tasks. ACT, ManiFlow, and A-FAR report mean ± standard deviation over three training seeds; $\pi_{0.5}$ is reported from one multi-task training run. Higher is better.

	Clean		Stage Occlusion		Random-time Occlusion	
Method	SR (%) ↑	TPS (%) ↑	SR (%) ↑	TPS (%) ↑	SR (%) ↑	TPS (%) ↑
ACT	10.3±1.2	46.2±6.8	24.7±2.3	58.1±1.7	15.7±3.2	46.6±0.4
$\pi_{0.5}$	13.0	55.2	42.0	73.4	4.0	40.8
ManiFlow	45.7±1.5	70.5±6.4	60.7±3.1	81.5±3.2	15.0±3.5	50.5±2.7
A-FAR	**51.3±4.2**	**73.9±5.4**	**63.7±2.1**	**83.9±0.9**	**18.7±1.2**	**51.9±1.4**

5.2 Main Results

Overall task performance. As shown in Table 1, A-FAR achieves the highest macro-averaged SR and TPS under all three visibility conditions. Compared with its direct ManiFlow backbone, A-FAR improves SR by 5.6, 3.0, and 3.7 pp under Clean, Stage Occlusion, and Random-time Occlusion, respectively, while consistently improving TPS. The gain under Clean indicates that the additional active-view modeling does not trade off nominal manipulation capability, while the improvement under Stage Occlusion shows that A-FAR better exploits the visibility-recovery behavior represented in the demonstrations.

Robustness to temporally unexpected occlusions. Random-time Occlusion introduces a stronger distribution shift by disrupting visibility at timings not aligned with the demonstration stages. Under this condition, A-FAR improves SR from 15.0% to 18.7% over ManiFlow, a 24.7% relative improvement, and increases TPS from 50.5% to 51.9%. The SR improvement is observed across all five tasks, suggesting that the gain is not driven by a single manipulation behavior. In comparison, $\pi_{0.5}$ reaches 42.0% SR under the training-matched Stage condition but drops to 4.0% under Random-time Occlusion, highlighting the difficulty of maintaining manipulation performance when visibility failures occur at unexpected execution stages.

5.3 Active-View Recovery Analysis

Visibility acquisition under unexpected occlusions. Table 2 reveals a complementary aspect of active-view behavior that is not captured by task success alone. While A-FAR improves average CVG from 28.57 pp to 29.64 pp over its ManiFlow backbone, $\pi_{0.5}$ achieves the highest CVG on all five tasks, with an average of 34.42 pp. Qualitative inspection of the evaluation rollouts is consistent with this difference: following randomly timed occlusions, $\pi_{0.5}$ more frequently exhibits camera responses that restore visibility of task-relevant regions, although recovery is not successful in every episode. This result demonstrates that CVG captures a distinct capability from SR and TPS—namely, whether the policy actively acquires additional visual information when its current observation becomes insufficient.

From visibility recovery to effective manipulation. Strong visibility acquisition, however, does not necessarily translate into successful task execution. The contrast between $\pi_{0.5}$'s high CVG and its lower Random-time SR exposes an *acquisition–exploitation gap*: an active-view policy must not only recover an informative observation, but also incorporate the recovered information into subsequent closed-loop manipulation. One possible explanation is that viewpoint recovery and precise manipulation place different demands on the policy; recovering task-relevant content can

Table 2: **Active-view analysis under Random-time Occlusion.** Each entry reports **CVG / Joint**. CVG measures camera-attributable visibility recovery (pp), while $\text{Joint} = \text{SR} \times \text{CVG}/100$ jointly reflects visibility recovery and task success. Higher is better.

Task	ACT	$\pi_{0.5}$	ManiFlow	**A-FAR**
Cube	16.49/0.00	**32.81**/0.00	26.25/2.18	25.88/**3.44**
Drawer	19.71/8.87	**35.65**/0.00	30.47/11.67	32.27/**12.91**
Slide	17.64/**5.29**	**28.72**/2.87	22.50/2.99	24.08/4.02
Stack	20.36/0.00	**32.94**/0.00	28.66/0.00	28.93/**0.49**
Sweep	21.49/0.71	**41.99**/4.20	34.97/5.25	37.04/**8.04**
Avg.	19.14/2.97	**34.42**/1.41	28.57/4.42	29.64/**5.78**

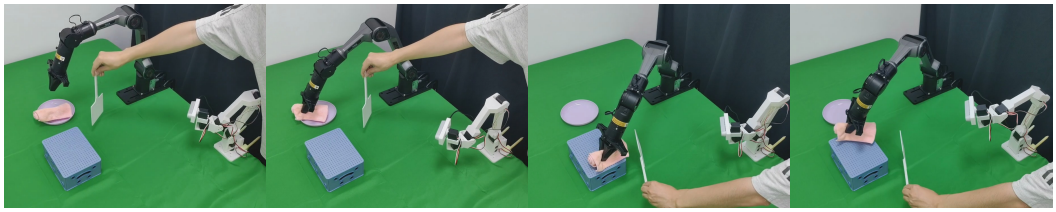

Figure 4: **Real-world deployment of A-FAR.** Representative rollout under an externally introduced occlusion. From left to right, task-relevant visibility is disrupted during execution, the active camera changes viewpoint to recover the scene, and manipulation proceeds after the target region becomes observable again. The sequence demonstrates closed-loop active-view behavior on our physical manipulation platform.

benefit from broad visual-semantic priors, whereas exploiting the recovered view requires it to be consistently aligned with the ongoing manipulation state.

To summarize this coupling, we additionally report the parameter-free SR–CVG product in Table 2. A-FAR achieves the highest Joint score on four of the five tasks and the highest average score (5.78), compared with 4.42 for ManiFlow, 2.97 for ACT, and 1.41 for $\pi_{0.5}$. Together with the raw CVG results, this suggests that A-FAR does not simply maximize visibility gain; rather, it more effectively couples active visibility recovery with downstream manipulation.

5.4 Ablation Studies

Effect of future modeling and viewpoint stabilization. Future relational modeling provides a consistent improvement in the three-condition average, increasing SR from 41.0% for ManiFlow to 42.3% with future-evolution supervision, 43.0% with feature fusion, and 44.7% for the full model. A substantially larger effect comes from viewpoint stabilization: expressing the same observations in the moving camera frame reduces average SR from 41.0% to 28.0% for ManiFlow and from 44.7% to 27.0% for A-FAR. Notably, A-FAR's advantage disappears without the shared base frame, indicating that viewpoint-stable geometry is a prerequisite for effectively exploiting temporal relational cues.

Table 3: **Ablation study.** SR (%) under Clean (C), Stage (S), and Random-time (R) Occlusion.

Variant	C	S	R	Avg.
ManiFlow (base)	44	60	15	41.0
+ Future Evo.	47	64	16	42.3
+ Future Fusion	47	63	18	43.0
A-FAR (base)	**50**	**66**	**19**	**44.7**
ManiFlow (camera)	26	44	14	28.0
A-FAR (camera)	22	43	16	27.0

5.5 Real-World Feasibility

We further deploy A-FAR on our real-world active-camera manipulation platform to evaluate whether the complete perception–control pipeline can operate under physical visibility disruptions. As illustrated in Fig. 4, when an external occluder blocks task-relevant visual information during execution, the observation arm changes its viewpoint to recover the scene and the policy subsequently continues manipulation. These rollouts demonstrate that the learned active-view behavior can be executed on physical hardware without introducing a separate perception or view-planning module.

6 Conclusion

We study physical active vision as closed-loop recovery from visibility failures during manipulation. We introduce BAVO-Bench, which evaluates active-view recovery under Clean, Stage Occlusion, and temporally unexpected Random-time Occlusion, and propose A-FAR, which combines base-frame 3D observations with 4D relational distillation for viewpoint-stable and future-aware control. Across five bimanual tasks, A-FAR consistently improves over its ManiFlow backbone, including under Random-time Occlusion, and transfers to our real-world active-camera platform. Our results further show that recovering visibility does not necessarily translate into task success, highlighting a remaining challenge in how policies exploit newly acquired observations for subsequent closed-loop control.

AI USE STATEMENT

Generative AI tools were used to provide feedback on research methodology and experimental design, and to assist with literature search and summarization, code development, and the drafting and editing of the manuscript. All research decisions, experimental implementations, results, and claims were independently reviewed and verified by the authors. The authors take full responsibility for the final content of this work, including any text, code, or artifacts produced with the aid of generative AI.

REFERENCES

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025b. doi: 10.15607/RSS.2025.XXI.010.

Ricson Cheng, Arpit Agarwal, and Katerina Fragkiadaki. Reinforcement learning of active vision for manipulating objects under occlusions. In *Proceedings of the 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pp. 422–431. PMLR, 2018. URL https://proceedings.mlr.press/v87/cheng18a.html.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.026.

Daesol Cho, Youngseok Jang, Danfei Xu, and Sehoon Ha. Egoavflow: Robot policy learning with active vision from human egocentric videos via 3d flow, 2026. URL https://arxiv.org/abs/2602.22461.

Ian Chuang, Andrew Lee, Dechen Gao, M-Mahdi Naddaf-Sh, and Iman Soltani. Active vision might be all you need: Exploring active vision in bimanual robotic manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7952–7959. IEEE, 2025.

Yuxin He, Ruihao Zhang, Tianao Shen, Cheng Liu, and Qiang Nie. Towards exploratory and focused manipulation with bimanual active perception: A new problem, benchmark and strategy, 2026. URL https://arxiv.org/abs/2602.01939.

Hyeongjun Heo, Seungyeon Woo, Sang Min Kim, Junho Kim, Junho Lee, Yonghyeon Lee, and Young Min Kim. Anycamvla: Zero-shot camera adaptation for viewpoint robust vision-language-action models. *arXiv preprint arXiv:2603.05868*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R 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. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of

Proceedings of Machine Learning Research, pp. 2679–2713. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Jiaxin Li and RHOS Team. OpenD4RT: An unofficial PyTorch/GPU implementation of D4RT for 4D reconstruction and tracking. https://github.com/Lijiaxin0111/Open-d4rt, 2026. Accessed: 2026-09-15.

Yibo Li, Enshen Zhou, Rui Chen, Yanjun Ding, Mengzhen Liu, Yi Han, Jiabo Zhan, Lipeng Wang, Shanghang Zhang, and Lu Sheng. Activearena: Benchmarking and understanding active perception in robotic manipulation. *arXiv preprint arXiv:2609.24124*, 2026.

Mengzhen Liu, Enshen Zhou, Cheng Chi, Yi Han, Shanyu Rong, Liming Chen, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Sapave: Towards active perception and manipulation in vision-language action models for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 37164–37174, June 2026a.

Zhenyang Liu, Yongchong Gu, Yikai Wang, Xiangyang Xue, and Yanwei Fu. Activevla: Injecting active perception into vision-language-action models for precise 3d robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8141–8151, June 2026b.

Giacomo Spigler. TAVIS: A benchmark for egocentric active vision and anticipatory gaze in imitation learning. *arXiv preprint arXiv:2605.07943*, 2026. URL https://arxiv.org/abs/2605.07943.

Guokang Wang, Hang Li, Shuyuan Zhang, Di Guo, Yanhong Liu, and Huaping Liu. Observe then act: Asynchronous active vision-action model for robotic manipulation. *IEEE Robotics and Automation Letters*, 10(4):3422–3429, 2025. doi: 10.1109/LRA.2025.3541334.

Haoyu Xiong, Xiaomeng Xu, Jimmy Wu, Yifan Hou, Jeannette Bohg, and Shuran Song. Vision in action: Learning active perception from human demonstrations. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 5450–5463. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/xiong25a.html.

Ge Yan, Jiyue Zhu, Yuquan Deng, Shiqi Yang, Ri-Zhao Qiu, Xuxin Cheng, Marius Memmel, Ranjay Krishna, Ankit Goyal, Xiaolong Wang, and Dieter Fox. ManiFlow: A general robot manipulation policy via consistency flow training. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 2268–2293. PMLR, 2025. URL https://proceedings.mlr.press/v305/yan25a.html.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026. URL https://arxiv.org/abs/2602.15922.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently reconstructing dynamic scenes one D4RT at a time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7382–7392, June 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhang_Efficiently_Reconstructing_Dynamic_Scenes_One_D4RT_at_a_Time_CVPR_2026_paper.html.

Jianjun Zhang, Jian Zhu, Taiyi Su, Chong Ma, Zitai Huang, Yi Xu, and Hanli Wang. Learning 4D geometric priors for inference-efficient world action models. *arXiv preprint arXiv:2607.05468*, 2026b. URL https://arxiv.org/abs/2607.05468.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016.

Shuai Zhou, Kaisheng Pang, Wenxuan Song, Wenjie Zhang, Xinhu Zheng, and Haoang Li. Activescale: Scaling active perception for robots across model, data, and hardware. *arXiv preprint arXiv:2609.18514*, 2026.

A BAVO-Bench Details

A.1 Task and Milestone Definitions

BAVO-Bench consists of five multi-stage bimanual manipulation tasks: *Cube Handoff*, *Block Stacking*, *Drawer Stowing*, *Slide Retrieval*, and *Dustpan Sweeping*. Each task is decomposed into five ordered milestones that describe meaningful progress toward task completion. Milestones are determined from privileged simulator states and contact predicates for evaluation only. The fifth milestone also defines task success in all tasks. Table 4 summarizes the task objectives and milestone progression.

A.2 Occlusion Protocol

Both occlusion conditions introduce external visibility disruptions while keeping the underlying manipulation task unchanged. The occluder remains fixed in the world after each placement; consequently, recovering visibility requires the policy to actively move the observation camera.

Stage Occlusion. Stage Occlusion associates each task with three task-relevant occlusion targets. The first occluder is placed at episode reset, and the target is updated at two predefined task-stage transitions. The resulting intervention schedule is summarized in Table 5. Importantly, the target is determined directly by the current task stage rather than inferred online from the policy behavior.

Random-time Occlusion. Random-time Occlusion uses the same stage-dependent target sequence but decouples subsequent interventions from task transitions. In addition to the initial occlusion at reset, one trigger step is sampled at the beginning of each episode from each of six predefined control-step windows:

$$[60, 130],\ [130, 200],\ [200, 270],\ [270, 340],\ [340, 410],\ [410, 480],$$

subject to consecutive triggers being separated by more than 60 steps. When an intervention is triggered, its target is selected according to the task stage reached at that time. Random-time therefore introduces visibility disruptions independently of the predefined stage transitions and may interrupt an ongoing manipulation behavior.

Occluder placement. Stage Occlusion and Random-time Occlusion use the same occlusion-generation procedure. For each event, candidate placements are generated between the active camera and the current task-relevant target, with the occluder oriented toward the camera-to-target viewing direction. A placement is accepted only when it substantially suppresses target visibility while maintaining clearance from the robot. The selected occluder then remains fixed until the next intervention.

A.3 Metric Implementation Details

Task progress. For TPS, milestone progress is recorded monotonically: once an episode reaches a milestone, subsequent state changes do not reduce its deepest recorded milestone d_i. As defined in Sec. 3.3, reaching M_5 also constitutes task success.

CVG evaluation. We report CVG only under Random-time Occlusion. For Eq. 2, we use a recovery horizon of $H = 60$ control steps, corresponding to 2.4 s at the 25 Hz control frequency. Random-time interventions are separated by more than 60 control steps, such that recovery windows from consecutive events do not overlap.

Table 4: **Task objectives and milestone progression in BAVO-Bench.** Each task contains five ordered milestones, with M_5 defining task success.

Task	Objective	Ordered milestones
Cube Handoff	Handoff a cube from the right arm to the left arm and place it on the target plate.	M_1: the right gripper contacts the cube; M_2: the right arm lifts the cube; M_3: both grippers hold the cube during handoff; M_4: the left gripper takes over and the right gripper releases; M_5: the left arm places the cube on the target plate.
Block Stacking	Move the base block to the central workspace and stack the top block on it.	M_1: the right gripper contacts the base block; M_2: the base block is placed in the stacking region and released; M_3: the left gripper contacts the top block; M_4: the top block is lifted into the pre-stacking region; M_5: a stable stack is formed with both grippers released.
Drawer Stowing	Open the drawer, stow an object inside, and close the drawer.	M_1: the right gripper contacts the drawer handle; M_2: the drawer is opened; M_3: the left gripper contacts the object; M_4: the object is placed and released inside the drawer; M_5: the drawer is closed with the object remaining inside.
Slide Retrieval	Slide the object to expose its handle, then retrieve it with the left arm.	M_1: the right gripper contacts the blade; M_2: the handle is exposed by sliding; M_3: the right gripper releases the object; M_4: the left gripper contacts the exposed handle; M_5: the left arm lifts the object from the support surface.
Dustpan Sweeping	Position the dustpan and sweep the target object into it.	M_1: the left gripper contacts the dustpan handle; M_2: the dustpan reaches a valid sweeping configuration; M_3: the right gripper contacts the broom; M_4: the broom contacts the target object during sweeping; M_5: the target object reaches the dustpan entrance.

Table 5: **Stage Occlusion schedule.** An initial occlusion is introduced at reset, followed by two stage-aligned target updates.

Task	Initial target	Transitions	Subsequent targets
Cube Handoff	cube	$M_2,\ M_4$	cube $\rightarrow$ plate
Block Stacking	base block	$M_2,\ M_4$	top block $\rightarrow$ stacking region
Drawer Stowing	drawer front	$M_2,\ M_3$	object $\rightarrow$ drawer front
Slide Retrieval	support platform	$M_1,\ M_2$	blade $\rightarrow$ handle
Dustpan Sweeping	dustpan	$M_2,\ M_3$	broom $\rightarrow$ target object

We retain only events for which the full recovery window is available. Events for which the episode terminates before the end of the window are excluded rather than padded. For an episode containing multiple retained events, we first average their event-level CVG values and then average across episodes, so episodes with more interventions do not receive greater weight.

If the target is not visible in the corresponding no-occluder rendering, i.e., $N_e^{\mathrm{noOcc}} = 0$, its normalized visibility is set to zero. Counterfactual views are rendered from the same simulator state as the policy rollout by temporarily changing only the evaluation camera pose and the presence of the benchmark occluder. These rendering changes never affect policy observations, actions, or environment dynamics.

A.4 Benchmark Rollout Visualizations

To further illustrate the benchmark settings, Figs. 5, 6, and 7 provide representative rollout visualizations for all five tasks under Clean, Stage Occlusion, and Random-time Occlusion, respectively. The tasks are arranged in the following order: *Cube Handoff*, *Block Stacking*, *Drawer Stowing*, *Slide Retrieval*, and *Dustpan Sweeping*. For each task, the first row shows the third-person scene view, while the second row shows the corresponding active-camera view. Frames are ordered from left to right to illustrate the temporal progression of each task.



Figure 5: **Rollout visualizations under Clean.** Representative sequences for the five benchmark tasks without external visibility intervention. For each task, the top row shows the third-person scene view and the bottom row shows the active-camera view. Frames are ordered from left to right.

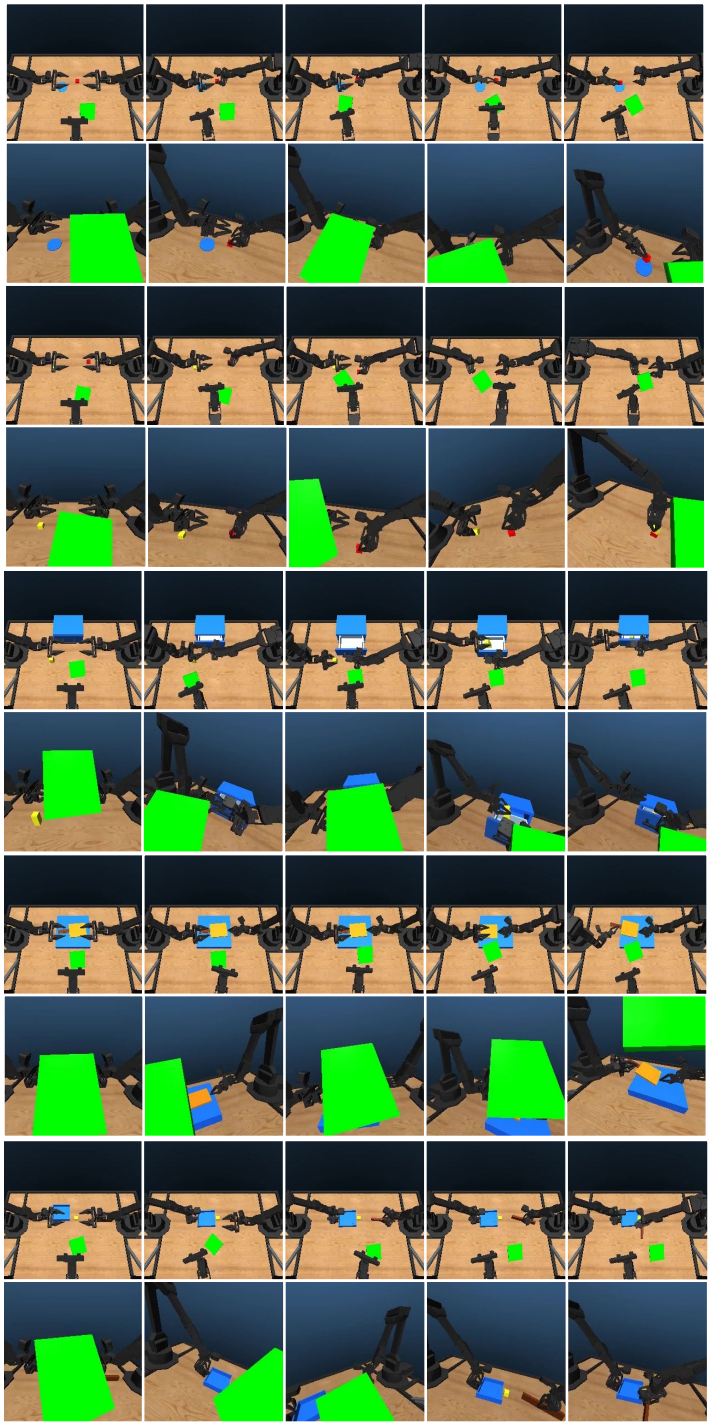

Figure 6: **Rollout visualizations under Stage Occlusion.** Representative sequences for the five benchmark tasks with stage-aligned occlusion events. For each task, the top row shows the third-person scene view and the bottom row shows the active-camera view. Frames are ordered from left to right. The sequences highlight how structured occlusions are introduced at predefined task transitions.

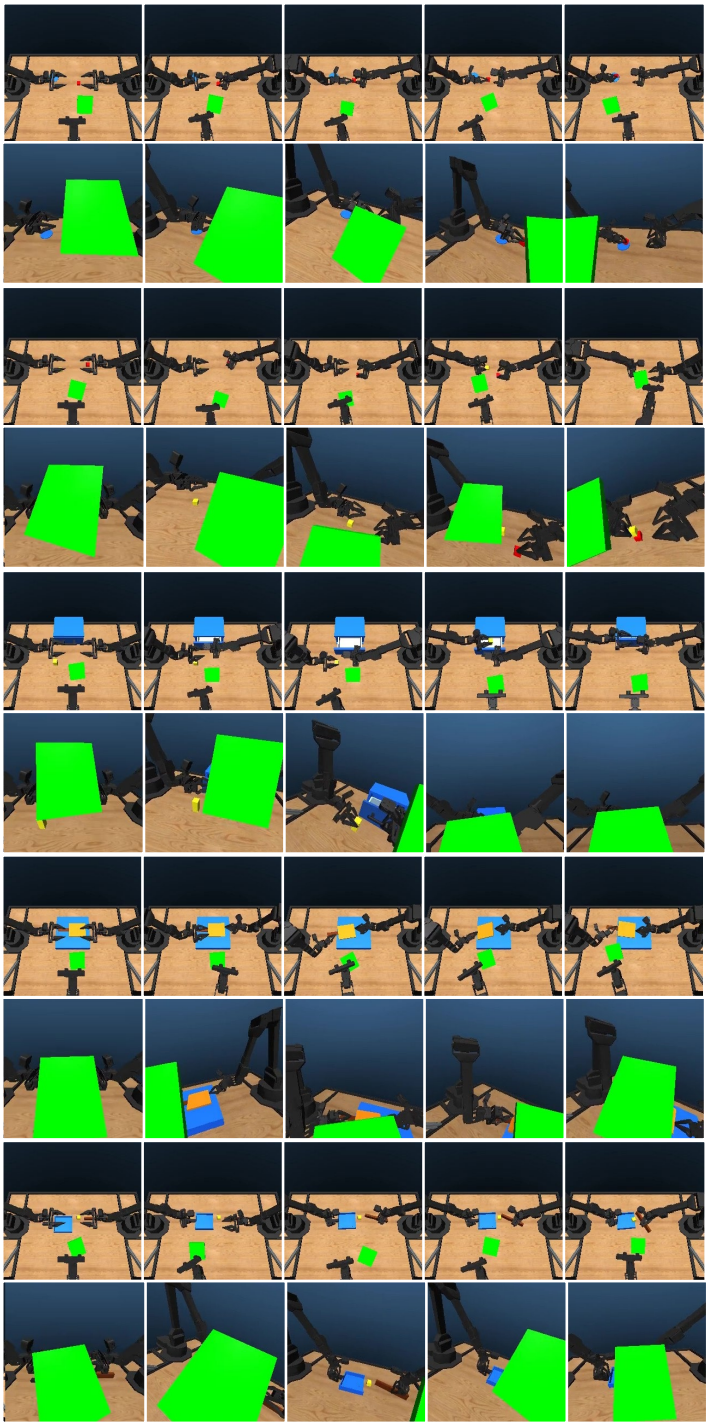

Figure 7: **Rollout visualizations under Random-time Occlusion.** Representative sequences for the five benchmark tasks with temporally unexpected occlusion events. For each task, the top row shows the third-person scene view and the bottom row shows the active-camera view. Frames are ordered from left to right. The sequences illustrate how visibility disruptions may occur during ongoing manipulation behaviors.

B A-FAR AND IMPLEMENTATION DETAILS

B.1 RELATIONAL DISTILLATION DETAILS

Teacher configuration. We use a frozen OpenD4RT teacher (Li & RHOS Team, 2026) with 32-frame RGB clips at 256×256 resolution. Using zero-based indexing, frame 16 is selected as the current anchor and frame 31 as the future target, corresponding to $\Delta t = 15$. For each sample, we query $N_q = 256$ points from the current frame. Following the notation in Eq. 3, the instantiated queries are

$$q_i^0 = (u_i, v_i, 16, 16, 16), \qquad q_i^F = (u_i, v_i, 16, 31, 31). \tag{7}$$

Teacher features are extracted from the final normalized decoder representation before the task-specific prediction heads. The resulting current relations R_t^T and relational evolution ΔR_t^T are precomputed and cached for policy training.

Dynamics-aware weighting. For each queried point, we compute a dynamics score from the magnitude of its teacher relational changes:

$$\begin{aligned} a_i &= \frac{1}{N_q - 1} \sum_{j \neq i} |\Delta R_{t,ij}^T|, \\ q_i &= \frac{N_q a_i}{\sum_k a_k}, \\ b_i &= 0.5 + 0.5 q_i, \qquad \omega_{ij} = \sqrt{b_i b_j}. \end{aligned} \tag{8}$$

Thus, q_i has unit mean across the queried points, while the uniform component in b_i preserves supervision for relations with weaker temporal variation.

The weighted norm in Eq. 6 is implemented as

$$\|A\|_{1,\omega} = \frac{\sum_{i<j} \omega_{ij} |A_{ij}|}{\sum_{i<j} \omega_{ij}}. \tag{9}$$

Dynamic weighting is applied only to $\mathcal{L}_{\text{evo}}$, while $\mathcal{L}_{\text{cur}}$ uses uniform weighting. The two terms contribute equally to the relational objective, with $\lambda_{\text{rel}} = 0.01$.

B.2 ARCHITECTURE AND TRAINING CONFIGURATION

Each observation is represented by 256 colored 3D points with 3D coordinates and RGB values. The ManiFlow point encoder produces a 128-dimensional feature for each point, while the 21-dimensional robot state is encoded into a 64-dimensional state representation.

WorldQueryFormer uses a hidden dimension of 128 and contains two transformer blocks with four attention heads per block. Each block consists of query self-attention, cross-attention to the current point features, and a feed-forward network. The encoded robot state is added to the initialization of the future queries, while the current point features serve directly as cross-attention memory.

Table 6 summarizes the main architecture and training settings.

C ADDITIONAL EXPERIMENTAL RESULTS

The main paper reports macro-averaged performance across the five BAVO-Bench tasks. In this section, we provide the complete task-level breakdown of success rate (SR) and task progress score (TPS) under Clean, Stage Occlusion, and Random-time Occlusion. We report the same four methods used in the main comparison: ACT, $\pi_{0.5}$, ManiFlow, and A-FAR. Here, $\pi_{0.5}$ refers to the monocular configuration used throughout the main experiments. Since Table 2 already reports task-level CVG and joint SR–CVG scores, we do not duplicate those results here.

C.1 TASK-LEVEL SUCCESS RATE

Table 7 reports the complete per-task SR results. The task-level breakdown further supports the robustness trend observed in the main results. In particular, under Random-time Occlusion, A-FAR

Configuration	Value
Point-cloud size	256
Teacher query points	256
Observation horizon	2
Action horizon	16
Point feature dimension	128
State feature dimension	64
WorldQueryFormer layers	2
Attention heads	4
Batch size	64
Optimizer	AdamW
Initial learning rate	1×10^{-4}
Training steps	100K
λ_{rel}	0.01

Table 6: **Main architecture and training settings for A-FAR.**

Table 7: **Task-level success rate (SR, %).** Results are reported under Clean (C), Stage Occlusion (S), and Random-time Occlusion (R). ACT, ManiFlow, and A-FAR are reported as mean±std over three training seeds; $\pi_{0.5}$ is evaluated from a single training run. Best performance for each task and condition is shown in bold.

Task	Method	C	S	R
Cube Handoff	ACT	0.0±0.0	6.7±7.6	0.0±0.0
	$\pi_{0.5}$	5.0	30.0	0.0
	ManiFlow	41.7±2.9	**68.3±5.8**	8.3±7.6
	A-FAR	**43.3±14.4**	58.3±2.9	**13.3±2.9**
Drawer Stowing	ACT	50.0±5.0	56.7±2.9	**45.0±8.7**
	$\pi_{0.5}$	25.0	75.0	0.0
	ManiFlow	**90.0±13.2**	**83.3±5.8**	38.3±14.4
	A-FAR	86.7±18.9	81.7±5.8	40.0±18.0
Slide Retrieval	ACT	0.0±0.0	31.7±15.3	**30.0±8.7**
	$\pi_{0.5}$	0.0	55.0	10.0
	ManiFlow	33.3±12.6	58.3±2.9	13.3±2.9
	A-FAR	**41.7±17.6**	**65.0±8.7**	16.7±2.9
Block Stacking	ACT	0.0±0.0	0.0±0.0	0.0±0.0
	$\pi_{0.5}$	10.0	15.0	0.0
	ManiFlow	8.3±7.6	13.3±2.9	0.0±0.0
	A-FAR	**13.3±11.5**	**23.3±10.4**	**1.7±2.9**
Dustpan Sweeping	ACT	1.7±2.9	28.3±16.1	3.3±5.8
	$\pi_{0.5}$	25.0	35.0	10.0
	ManiFlow	55.0±15.0	80.0±8.7	15.0±5.0
	A-FAR	**71.7±7.6**	**90.0±5.0**	**21.7±7.6**

improves over its ManiFlow backbone on all five tasks, despite substantial differences in task difficulty. The gains are especially visible on Cube, Slide, and Sweep, while Stack remains challenging for all evaluated methods.

C.2 Task-Level Progress

Table 8 provides the corresponding task-level TPS. Compared with binary success, TPS reveals partial task completion when a policy fails to reach the final goal. Under Random-time Occlusion, A-FAR improves TPS over ManiFlow on four of the five tasks, while the two methods remain close on Stack. Together with the SR results, this shows that the aggregate improvements reported in the main paper are distributed across multiple manipulation tasks rather than being dominated by a single task.

Table 8: **Task-level task progress score (TPS, %).** Results are reported under Clean (C), Stage Occlusion (S), and Random-time Occlusion (R). ACT, ManiFlow, and A-FAR are reported as mean±std over three training seeds; $\pi_{0.5}$ is evaluated from a single training run. Best performance for each task and condition is shown in bold.

Task	Method	C	S	R
Cube Handoff	ACT	33.0±2.6	44.0±3.6	24.0±6.2
	$\pi_{0.5}$	45.0	68.0	31.0
	ManiFlow	**70.3±5.0**	**87.7±2.3**	45.3±5.7
	A-FAR	69.7±13.2	84.0±1.7	**47.7±6.0**
Drawer Stowing	ACT	78.7±3.1	80.7±2.1	**77.3±4.7**
	$\pi_{0.5}$	70.0	90.0	51.0
	ManiFlow	**96.0±5.3**	**93.3±2.3**	65.3±5.5
	A-FAR	94.7±7.6	92.7±1.2	66.0±9.6
Slide Retrieval	ACT	48.0±19.0	65.7±9.5	**62.7±3.8**
	$\pi_{0.5}$	40.0	74.0	53.0
	ManiFlow	63.7±7.0	74.3±2.5	58.0±3.5
	A-FAR	**67.7±7.6**	**77.3±7.0**	59.7±4.2
Block Stacking	ACT	24.3±3.5	28.7±2.9	22.3±2.1
	$\pi_{0.5}$	**69.0**	69.0	22.0
	ManiFlow	44.7±20.3	62.7±6.7	**30.3±3.2**
	A-FAR	53.7±20.6	**69.7±4.0**	28.7±2.1
Dustpan Sweeping	ACT	47.0±17.3	71.3±8.6	46.7±2.5
	$\pi_{0.5}$	52.0	66.0	47.0
	ManiFlow	77.7±3.1	89.7±5.0	53.7±6.5
	A-FAR	**84.0±5.3**	**96.0±2.0**	**57.3±5.1**

D Real-World Active-Vision Platform

We use a CamBot equipped with an Intel Realsense D435i camera as our observation arm and a Piper arm as our manipulation arm. The two arms are mounted along the edge of a table and oriented parallel to each other, as shown in Fig. 8.

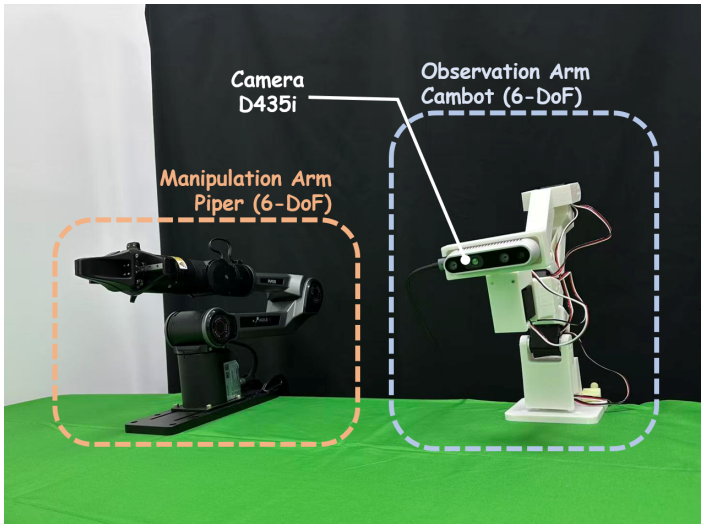

Figure 8: **Real-world Experiement Setup** At the home position, the manipulation arm's gripper points toward the workspace, and the camera is oriented in the same direction.