AWAM: ACTIVE TRAJECTORY-CONDITIONED VIEW IMAGINATION FOR ROBUST ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

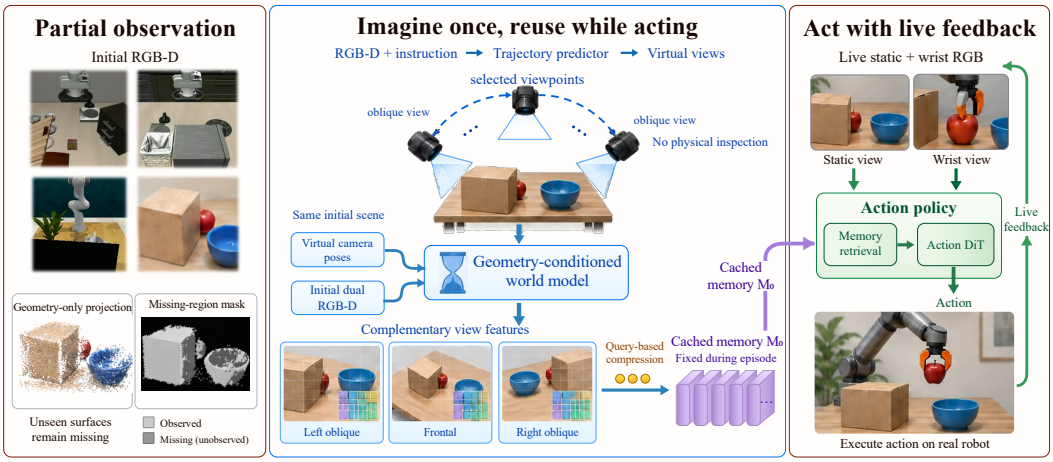

Figure 1: Overview of AWAM. From partially observed initial RGB-D inputs, AWAM imagines task-conditioned virtual views and compresses them into a predictive memory once per episode. The policy then reuses this memory together with live visual observations for closed-loop manipulation, without physically executing the predicted camera trajectory.

ABSTRACT

Rapid progress in embodied intelligence has enabled increasingly capable robots to follow language instructions and manipulate diverse objects. However, strong results in structured environments do not automatically transfer to open-world conditions, where clutter, occlusion, and changing viewpoints can obscure the spatial relationships required for action. To address this perceptual gap, a virtual active-perception paradigm is introduced. AWAM, an Active World-Action Model actively plans where to observe and imagines what will be seen along the planned trajectory before acting. Task-related virtual camera motion is predicted and used as a structured query into a geometry-conditioned world model, producing a reusable spatial prior for manipulation, which is later compressed into a fixed memory and fused with live observations through cross-attention for diffusion-based action prediction. AWAM achieves competitive success rates with 75.9% on LIBERO-Occ, 62.4% on LIBERO-Plus, 76.5% on RLBench-OG Occlusion1, and 94.4% in real-robot trials. These results support improved robustness to the evaluated occlusions and distribution shifts, addressing part of the gap between structured manipulation and less controlled environments.

1 INTRODUCTION

Embodied intelligence has advanced rapidly through large-scale robot learning, vision–language pretraining, and generative action modeling. Increasingly capable policies can translate language goals and visual observations into manipulation behaviors across diverse tasks (Brohan et al., 2022; 2023; Ghosh et al., 2024; Kim et al., 2024; Black et al., 2025; Chi et al., 2025).

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.

However, much of this progress is evaluated in relatively structured settings in which camera configurations, scene layouts, and task-relevant objects remain sufficiently observable. Occlusion-focused and distribution-shift benchmarks expose a complementary limitation: once deployment becomes less structured, an unfavorable camera pose, clutter, or a shifted layout can hide the target or interaction region and remove the evidence required for control (Li et al., 2026; Fei et al., 2026; Bai et al., 2026). This creates a perceptual bottleneck rather than only an action-modeling bottleneck. More expressive action models alone cannot directly recover scene evidence that is absent or highly ambiguous in the input, so performance gains obtained under well-observed scenes may not transfer to open-world conditions with incomplete visibility.

Several lines of work have begun to address this problem. Physical active perception acquires additional evidence by moving the camera or robot, but introduces extra execution steps and can alter the robot or scene state. Virtual-view methods avoid this cost by selecting or synthesizing complementary observations. Candidate-view approaches choose from predefined cameras (Liu et al., 2026; Xu et al., 2026); TVVE predicts task-conditioned virtual poses and re-renders reconstructed geometry (Bai et al., 2026); and world-action models such as Motus and Fast-WAM exploit predictive visual modeling to strengthen action representations (Bi et al., 2026; Yuan et al., 2026). These directions establish the value of additional visual context, yet the underlying limitation is not fully resolved: virtual perception is often restricted to isolated views, reconstruction-based rendering cannot reveal surfaces absent from the initial geometry, and predictive world models are commonly organized around action-induced futures rather than deliberate viewpoint queries.

To address these questions, task-conditioned virtual active perception is formulated as trajectory-conditioned view imagination. Instead of selecting a single view or a small set of independent poses, a virtual camera trajectory specifies how perception should evolve to expose task-relevant evidence. Such a trajectory can reorient toward a target, approach fine interaction geometry, or move around an occluder when no single viewpoint is sufficient. Importantly, the predicted trajectory is never physically executed; it serves only as a sequence of virtual queries, thus retrieving the available evidence and learned spatial priors into a representation that better supports control, separating perceptual exploration from robot motion.

Based on this formulation, AWAM is introduced as an Active World-Action Model that realizes the virtual-perception process before closed-loop control. A task-conditioned trajectory predictor first produces virtual camera poses for better observation from the initial RGB-D image and language instruction. The calibrated observations are then back-projected into coarse 3D geometry and rendered along the predicted path. A geometry-conditioned world model fills unsupported regions and produces multi-view latent features, which are compressed into a persistent world memory. During manipulation, live static- and wrist-camera observations retrieve relevant predictive context from this memory, and a diffusion policy generates short-horizon action chunks in closed loop. The resulting design uses world modeling as an internal perceptual mechanism for asking where to look and what would be seen, rather than as a prediction of the robot's executed future.

Evaluation is conducted across three simulation benchmarks and real-world manipulation. Success reaches 75.9% on LIBERO-Occ, 62.4% on LIBERO-Plus, $76.5 \pm 1.1\%$ on RLBench-OG Occlusion1, and 94.4% in real-robot trials. On LIBERO-Occ, removing the entire predictive-memory branch reduces average success by 11.5 percentage points, and replacing learned task-conditioned virtual trajectory with sampled viewpoints reduces it by 5.7 points. Together, these results demonstrate the effectiveness of learned virtual camera motion and persistent predictive memory for manipulation under incomplete visibility. The contributions are threefold:

- A trajectory-conditioned virtual active-perception paradigm is introduced for manipulation, replacing isolated or physically executed viewpoint acquisition with a continuous sequence of task-directed virtual queries.
- AWAM is proposed to transform the predicted observation trajectory into geometry-consistent multi-view representations and compress them into persistent world memory as a reusable spatial prior for manipulation under partial observability.
- Extensive evaluations across three challenging simulation benchmarks and real-world manipulation demonstrate strong performance under occlusion and distribution shifts and confirm the benefits of learned camera motion and predictive memory through ablations.

2 Related Work

Robot learning and action generation. Large-scale robot policies map visual observations and language instructions directly to actions. Recent VLA models and diffusion-based policies improve generalization, action representation, and efficient decoding through large-scale pretraining and expressive generative objectives (Brohan et al., 2022; 2023; Ghosh et al., 2024; Kim et al., 2024; Black et al., 2025; Chi et al., 2025; Pertsch et al., 2025; Kim et al., 2025). These methods primarily focus on how actions are represented and learned from available observations. In contrast, AWAM targets the perceptual evidence supplied to the policy, constructing additional task-conditioned spatial context when relevant scene information is poorly observed or occluded.

Multiview manipulation under occlusion. Occlusion can be mitigated by acquiring or synthesizing complementary visual evidence through multiview perception, viewpoint selection, and imagined views (Liu et al., 2026; Bai et al., 2026; Li et al., 2026; Xu et al., 2026). Existing approaches typically rely on additional sensing, reconstructed views, or a small fixed number of generated observations. AWAM instead predicts a task-conditioned virtual camera trajectory and consolidates the generated multi-view sequence into persistent world memory for closed-loop manipulation. This formulation turns occlusion handling into trajectory-conditioned predictive perception, where the model jointly reasons about where to observe, what would be observed, and how this information should support subsequent control.

World models for robotic manipulation. Recent video diffusion and world models have shown strong capability in modeling long-horizon visual dynamics, while text-, action-, and camera-conditioned variants further enable controllable prediction of future or novel observations (Zhang et al., 2025a; 2026). Predictive world models provide representations of how visual observations may evolve and have increasingly been integrated with robot control. Video Prediction Policy (Hu et al., 2025) conditions control on features learned from predictive video modeling. Motus (Bi et al., 2026) unifies visual understanding, video generation, inverse dynamics, and action prediction within a latent world-action model. Fast-WAM (Yuan et al., 2026) separates the benefits of video co-training from explicit test-time future generation and shows that strong action prediction can be retained without iterative future-video rollout. Unlike approaches centered on predicting the future scene induced by robot actions, AWAM uses world modeling as a form of active perception. A task-conditioned camera trajectory specifies where the current scene should be queried, and a geometry-conditioned world model predicts visual representations along this trajectory. These representations are consolidated into a compact world memory and reused throughout manipulation rather than interpreted as the robot's executed future.

3 Method

3.1 Problem Formulation and Overview

We consider language-conditioned manipulation with a static camera s and a wrist-mounted camera w. At control step t, the policy observes RGB images I_t^s and I_t^w, proprioceptive state s_t, and language instruction ℓ. At initialization it also receives metric depth and calibration:

$$\mathcal{O}_0 = \left(\{I_0^v, D_0^v, K^v, T_0^v\}_{v\in\{s,w\}}, s_0, \ell\right), \tag{1}$$

where K^v denotes camera intrinsics and $T_0^v \in \mathrm{SE}(3)$ maps camera coordinates to the world frame. Depth and calibration are used to build the initial memory; the action policy consumes live RGB observations and proprioception. AWAM separates memory construction from repeated action prediction (Fig. 2): given the initial observation, AWAM constructs an active perception memory and subsequently conditions action prediction on both the memory and the current sensory evidence:

$$\widehat{\mathcal{T}} = G_\phi(I_0^s, D_0^s, \ell; T_0^w), \tag{2}$$

$$M_0 = \mathcal{R}_\eta\left(\mathcal{F}_\psi(\mathcal{O}_0, \widehat{\mathcal{T}})\right), \tag{3}$$

$$A_t \sim \pi_\theta(\cdot \mid I_t^s, I_t^w, s_t, \ell, M_0). \tag{4}$$

Here, G_ϕ predicts task-relevant virtual viewpoints which may offer clearer observations, $\mathcal{F}_\psi$ extracts geometry-conditioned predictive representations along the planned perception trajectory, and $\mathcal{R}_\eta$

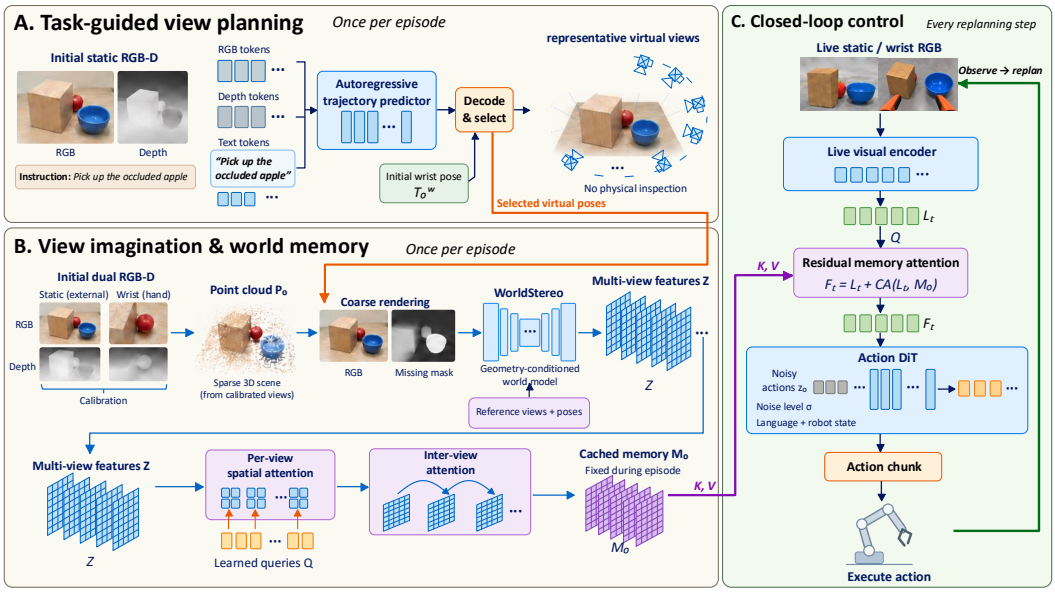

Figure 2: Overview of AWAM. A task-guided virtual camera trajectory guides a geometry-conditioned world model to construct predictive memory once per episode. During closed-loop control, the cached world memory is fused with live visual observations to predict action chunks.

compresses them into a compact world memory. The action policy π_θ then combines this memory with live observations during closed-loop control.

3.2 Task-Conditioned Virtual View Planning

An unfavorable initial wrist-camera pose can hide the object needed for manipulation, leaving the policy to act on incomplete visual evidence. Rather than relying on the initial viewpoint alone, a task-conditioned camera trajectory is predicted to specify where additional virtual observations should be queried for better scene perception.

Demonstrated wrist-camera paths provide proxy supervision for task-relevant virtual queries. Each trajectory is parameterized relative to its initial wrist-camera pose, making prediction invariant to the global scene origin. For a wrist pose T_i^w, the relative transform is defined as $T_i = (T_0^w)^{-1} T_i^w$.

The relative poses are discretized into trajectory tokens encoding translation, rotation, and trajectory scale. Initial RGB-D observations and the language instruction are encoded into a shared conditioning representation C. A transformer then autoregressively predicts the trajectory sequence:

$$\mathcal{L}_{\text{traj}} = -\mathbb{E}_{\mathcal{D}} \sum_{j=1}^{J} \log p_\phi(y_j \mid C, y_{<j}). \tag{5}$$

At inference, the decoded relative trajectory is transformed back into metric camera poses using the observed initial wrist pose T_0^w. A small set of representative viewpoints is retained along the trajectory rather than using every predicted pose. These virtual viewpoints define the camera queries passed to the world model and provide diverse observations of task-relevant regions while avoiding redundant predictions from nearby poses.

3.3 Geometry-Conditioned World Memory

A virtual view contains both geometrically supported pixels and regions that the initial cameras have not observed. We first unproject initial RGB-D observations into a colored point cloud,

$$\mathcal{P}_0 = \bigcup_{v \in \{s,w\}} \text{BackProject}(I_0^v, D_0^v, K^v, T_0^v). \tag{6}$$

Rendering $\mathcal{P}_0$ at the predicted camera poses preserves evidence from observed surfaces, but leaves holes where the reconstruction has no support. Such renderings become increasingly incomplete under viewpoint changes, since surfaces that are occluded or outside the initial fields of view have no geometric support, often resulting in large holes and unreliable visual features.

WorldStereo (Zhang et al., 2026) is employed as a geometry-conditioned world model and adapted to manipulation scenes. Intermediate geometry-conditioned features are extracted instead of passing generated RGB video to the action model, following the predictive-representation perspective of Hu et al. (2025). The generation target is an alternate view of the initial scene: objects and the robot are intended to remain fixed while the virtual camera changes pose.

The memory encoder compresses the predicted sequence. It aggregates spatial features within each predicted view and subsequently exchanges information across virtual viewpoints through inter-view attention. The resulting fixed-size token set summarizes the trajectory-level visual prediction while remaining efficient to retrieve during control, capturing what the robot is expected to observe from task-relevant viewpoints while remaining grounded in the geometry of the current scene.

3.4 Memory-Conditioned Closed Loop Control

At each control step, the current static- and wrist-camera observations are encoded into live visual tokens L_t. Predictive context is retrieved from the cached world memory M_0 using residual cross-attention,

$$F_t = L_t + \mathrm{CA}(L_t, M_0), \tag{7}$$

where L_t serves as the attention query and M_0 provides the keys and values. Although the memory remains fixed throughout the episode, its retrieved content changes with the current observation. This allows the policy to access different parts of the predictive context as the manipulation state evolves, while the residual connection preserves a direct pathway from live perception to action prediction.

A conditional diffusion transformer is used to generate short-horizon action chunks. Given the fused visual representation F_t, proprioceptive state s_t, and language instruction ℓ, the policy predicts

$$\widehat{A}_t = \mathcal{D}_\theta\left(z_\sigma, \sigma; F_t, s_t, \ell\right), \tag{8}$$

where z_σ denotes the noisy action representation at diffusion level σ. The policy is trained on demonstrated action chunks using a standard diffusion denoising objective. During both action-policy training and evaluation, M_0 is constructed from the trajectory predicted by G_ϕ rather than from the demonstrated wrist-camera trajectory, avoiding a train–test mismatch in the memory input.

During closed-loop execution, the geometry-conditioned memory is constructed once from the initial observation and retained throughout the episode. At each replanning step, the policy receives fresh visual and proprioceptive observations, retrieves relevant predictive context, and generates a new action chunk from the updated state. The cached memory therefore acts as persistent predictive context about task-relevant virtual viewpoints rather than as a continually updated scene representation. Live observations remain the primary source of information about the evolving manipulation state, while the memory provides complementary spatial evidence that may be unavailable from the current viewpoint.

4 Experiments

4.1 Experimental Setup

Data and Environments. Evaluation is conducted on three complementary simulation benchmarks. LIBERO-Occ (Li et al., 2026) tests robustness to scene-induced occlusion across the Spatial, Object, Goal, and Long suites; all 2,000 benchmark instances are evaluated, with 500 rollouts per suite. The Occlusion1 split of RLBench-OG (Bai et al., 2026), which contains ten occlusion-heavy manipulation tasks and uses a control interface different from LIBERO, provides a cross-benchmark test of the proposed world-memory mechanism. Both benchmarks directly test whether predicted viewpoints and imagined world memory predict task evidence omitted by the initial observation. Robustness beyond explicit occlusion is additionally evaluated on LIBERO-Plus (Fei et al., 2026),

Table 1: Success rate (%) on LIBERO-Occ. Each suite contains 500 evaluation rollouts.

Method	Spatial	Object	Goal	Long	Avg.
π_0	81.0	84.0	76.6	38.4	70.0
Motus	82.2	78.0	66.0	43.4	67.4
Fast-WAM	82.8	86.4	78.8	**44.2**	73.0
VIM	83.0	92.0	81.0	40.0	74.0
AWAM	**84.6**	**93.2**	**84.8**	41.0	**75.9**

which introduces controlled shifts in camera viewpoint, robot initial state, language instruction, illumination, background texture, sensor noise, and object layout. Environment-defined task success is reported for LIBERO-Occ and LIBERO-Plus, while mean $\pm$ standard deviation over three independent runs is reported for RLBench-OG. Real-world evaluation is conducted on an ARCHER-Y6 arm equipped with GP-100 grippers, scene-level and wrist-mounted Intel RealSense D435 cameras, and 100 original demonstrations.

Baselines. Baselines are selected according to the focus and evaluation ecosystem of each benchmark. For LIBERO-based evaluations, three categories are included: (i) representative vision-language-action policies such as π_0 (Black et al., 2025); (ii) world-model-augmented manipulation methods including Motus (Bi et al., 2026) and Fast-WAM (Yuan et al., 2026); and (iii) benchmark-specific occlusion methods represented by VIM (Li et al., 2026) on LIBERO-Occ. This set compares AWAM with both general-purpose policies and methods that explicitly introduce predictive or auxiliary visual representations. For RLBench-OG, the benchmark's evaluation ecosystem is followed using Diffusion Policy (Chi et al., 2025), ARP (Zhang et al., 2025b), RVT2 (Goyal et al., 2024), and TVVE (Bai et al., 2026). When released results do not cover the exact evaluation setting, the corresponding methods are reproduced and tuned using their recommended training recipes and evaluated under the same protocol as AWAM.

4.2 Experimental Results

Results on LIBERO-OCC. As shown in Table 1, AWAM achieves an average success rate of 75.9%. The improvement is consistent across three of the four suites: 84.6% on Spatial, 93.2% on Object, and 84.8% on Goal. They benefit from obtaining task-relevant evidence that is difficult to infer from the current view: the predicted viewpoint sequence queries the initially hidden region, while the world model converts those queries into a persistent representation that can be reused during manipulation. On Long, however, Fast-WAM obtains the highest success rate (44.2%), while AWAM reaches 41.0%. Long-horizon tasks remain challenging because the cached memory is constructed from the initial scene and is not regenerated after substantial object-state changes, which might be solved by regenerating memory each time when gripper state changes. Nevertheless, AWAM attains the highest overall average, indicating that active observations remain useful across a broad range of occluded manipulation problems.

Results on RLBench-OG. Table 2 presents the quantitative results on the Occlusion1 split of RLBench-OG. AWAM achieves an average success rate of 76.5%. The improvement is evident on tasks where the manipulation target or an interaction-relevant region is difficult to infer from the nominal observation. For example, AWAM reaches 94.7% success on water plants, improving over TVVE (88.0%), ARP (66.0%), and RVT2 (72.0%), and obtains 92.0% on scoop with spatula. It further achieves 100% success on five other tasks. Performance nevertheless remains low on block pyramid and solve puzzle. Both require precise multi-stage interaction, for which recovering visibility alone is insufficient. This suggests that predictive observation is most useful when partial observability is a primary bottleneck, but does not eliminate the intrinsic control difficulty of high-precision long-horizon tasks. Nevertheless, the overall result and the favorable average rank in Table 2 show that the benefit of active world memory extends across a diverse set of occlusion-sensitive tasks and transfers beyond the LIBERO environment.

Results on LIBERO-PLUS. As shown in Table 3, AWAM achieves an instance-weighted success rate of 62.4%, outperforming π_0 (53.6%), Fast-WAM (48.2%), and Motus (44.0%). The

Table 2: RLBench-OG Occlusion1 success (%, mean ± std. over three runs). Avg. Rank is averaged across 10 tasks; lower is better.

Method	Avg. ↑	Avg. Rank ↓	basketball in hoop	scoop with spatula	take plate off dish rack	water plants
Diffusion Policy	27.4±0.8	4.70	0.0±0.0	0.0±0.0	4.0±0.0	2.0±2.0
ARP	73.0±1.6	2.80	**100.0±0.0**	88.0±4.0	**100.0±0.0**	66.0±14.0
RVT2	72.8±2.0	3.05	**100.0±0.0**	82.7±7.5	98.7±1.9	72.0±8.6
TVVE	75.0±1.2	2.40	**100.0±0.0**	88.0±8.0	**100.0±0.0**	88.0±4.0
AWAM	**76.5±1.1**	**2.05**	**100.0±0.0**	**92.0±3.3**	**100.0±0.0**	**94.7±1.9**

Method	block pyramid	solve puzzle	take USB out of computer	close drawer	straighten rope	toilet seat down
Diffusion Policy	0.0±0.0	0.0±0.0	92.0±8.0	76.0±0.0	0.0±0.0	**100.0±0.0**
ARP	**14.0±2.0**	0.0±0.0	**100.0±0.0**	**100.0±0.0**	62.0±6.0	**100.0±0.0**
RVT2	4.0±3.3	0.0±0.0	**100.0±0.0**	**100.0±0.0**	**70.7±10.0**	**100.0±0.0**
TVVE	8.0±8.0	**2.0±2.0**	**100.0±0.0**	**100.0±0.0**	64.0±0.0	**100.0±0.0**
AWAM	12.0±3.3	1.3±1.9	**100.0±0.0**	**100.0±0.0**	65.3±3.8	**100.0±0.0**

Table 3: Success rate (%) on all 10,030 LIBERO-Plus instances; Total is instance-weighted.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Total
π_0	13.8	6.0	58.8	**85.0**	**81.4**	**79.0**	68.9	53.6
Motus	17.9	43.3	**74.1**	54.7	44.9	26.2	52.0	44.0
Fast-WAM	18.8	44.5	68.9	78.2	45.7	29.8	60.7	48.2
AWAM	**53.4**	**48.9**	61.0	84.3	61.1	60.7	**73.0**	**62.4**

gains are most pronounced under geometric perturbations that alter the observation configuration or scene structure, including camera viewpoint (53.4%), robot-state (48.9%), and object-layout shifts (73.0%). Robustness to viewpoint changes is under known calibration. This pattern is consistent with the design of AWAM: instead of relying solely on the current observation, the policy can leverage task-conditioned visual evidence predicted from multiple geometrically related viewpoints, reducing its sensitivity to changes in camera pose and scene geometry.

Real-world evaluation. AWAM is evaluated on three real-world manipulation tasks in which the initial observations provide incomplete task-relevant information: (i) retrieving a partially occluded apple from a box; (ii) picking a bottle from a cluttered tabletop and placing it into a nearby box; (iii) grasping a block initially occluded by a box and placing it into the box.

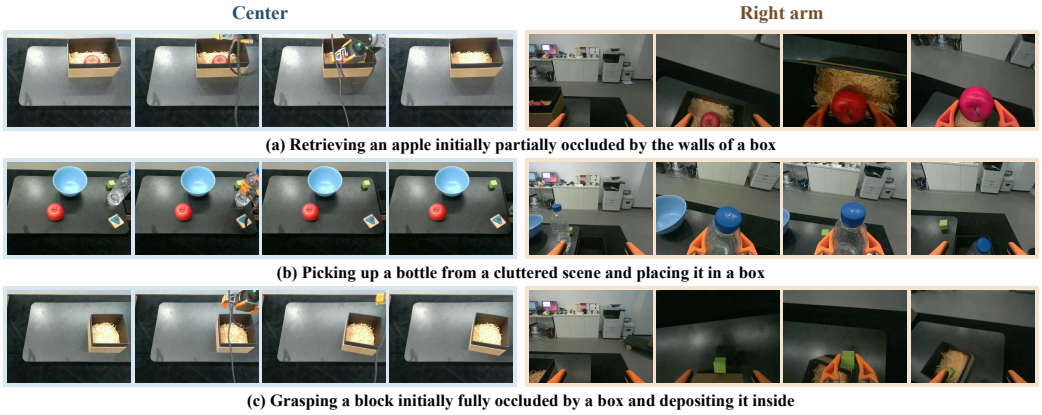

Figure 3: Real-world rollouts of AWAM under partial occlusion and clutters

As illustrated in Fig. 3, AWAM successfully handles partially occluded objects, cluttered scenes, and initially hidden targets in real-world grasping and placement. These rollouts highlight the practical value in addressing limited visibility and spatial ambiguity. Successful execution across these tasks

Table 4: Real-world manipulation, successes/trials (success rate, %).

Method	apple in box	bottle in cluster	block behind box	Overall
π_0	22/30(73.3%)	24/30(80.0%)	20/30(66.7%)	66/90(73.3%)
Motus	24/30(80.0%)	26/30(86.7%)	23/30(76.7%)	73/90(81.1%)
Fast-WAM	24/30(80.0%)	25/30(83.3%)	23/30(76.7%)	72/90(80.0%)
AWAM	**28/30 (93.3%)**	**30/30 (100.0%)**	**27/30 (90.0%)**	**85/90 (94.4%)**

Table 5: Component analysis on LIBERO-Occ. Δ denotes the change from full AWAM in percentage points.

Variant	Learned trajectory	Generation	Memory	Success (%) ↑	Δ
(a)Live observations only	×	×	×	64.4	-11.5
(b)Trajectory-conditioned live policy	✓	×	×	67.3	-8.6
(c)sampled spherical views	×	✓	✓	70.2	-5.7
(d)Learned path, warp only	✓	×	✓	69.0	-6.9
AWAM	✓	✓	✓	**75.9**	–

indicates that the active-perception memory framework extends beyond simulation to real-world robotic settings.

The robot platform and camera configuration described in Sec. 4.1 are used, with 30 trials per task for each method. As shown in Table 4, AWAM achieves 85/90 successful trials (94.4%), compared with 66/90 (73.3%) for π_0, 73/90 (81.1%) for Motus, and 72/90 (80.0%) for Fast-WAM. The consistent gains under different occlusion configurations suggest that imagining task-conditioned virtual viewpoints and caching them as predictive memory can provide complementary visual evidence for closed-loop manipulation without requiring additional physical camera motion.

4.3 Ablation Study And Qualitative Analysis

Component Analysis. To disentangle the contributions of each component, four controlled variants are evaluated while keeping the policy backbone, training data, and evaluation protocol unchanged. Results are summarized in Table 5.

Variant(a) shows that removing the entire predictive branch and relying only on live observations reduces the success rate from 75.9% to 64.4%, corresponding to the largest drop of 11.5 percentage points. This result confirms that the performance gain cannot be explained by the closed-loop policy alone, and that information constructed before control provides important complementary visual context when the current observations are incomplete or occluded. Variant(b), where the predicted camera trajectory is directly provided to the live policy without virtual-view generation or memory construction, performance improves to 67.3%. The 2.9-point gain over the live-only variant suggests that the task-conditioned trajectory itself carries a useful implicit manipulation prior. However, its substantial gap to the full model indicates that trajectory conditioning alone cannot substitute for explicit predictive visual context. Variant(c), which replaces the learned trajectory with spherical viewpoints uniformly sampled around the whole scene region achieves 70.2%. Despite preserving view diversity as well as the same generation and memory modules, the 5.7-point degradation shows that generic coverage of the scene is less effective than allocating virtual observations according to a learned, task-conditioned trajectory. Variant(d) retaining the learned trajectory while removing generative prediction and constructing memory only from warped geometric observations yields 69.0%. This variant preserves the views queried by the model, but can only reproject evidence already visible in the initial RGB-D observations. Its 6.9-point drop indicates that viewpoint selection alone is insufficient under severe occlusion, where previously unobserved surfaces cannot be recovered through geometric warping.

Taken together, these ablations reveal complementary roles of the proposed components. The predicted trajectory provides both task-dependent viewpoint selection and a weak planning prior, while generative prediction converts these queries into predictive visual context beyond the initially observed geometry. Compressing this predicted evidence into memory further allows the closed-loop

policy to access such information throughout manipulation. The full combination therefore achieves the highest success rate of 75.9% on LIBERO-Occ.

Sensitivity to Virtual-View Budget. The trajectory length K determines the amount of virtual visual context constructed before control. K varies in $\{7, 14, 21, 28\}$ while keeping all other settings fixed. Performance improves as the trajectory provides more virtual observations and saturates around $K = 21$, with success rates of 72.4%, 74.8%, 75.9% and 76.0% on LIBERO-Occ respectively. Increasing the length further to 28 brings no meaningful improvement, suggesting limited sensitivity to the exact choice of K once sufficient task-relevant scene coverage is obtained.

Predicted Virtual Views. For policy conditioning, intermediate predictive features are extracted after the first denoising update rather than completing the full RGB generation process. The RGB decodes in Fig. 4 are used only for qualitative diagnosis. Since AWAM uses a predicted camera trajectory to construct task-conditioned world memory, observations generated along this trajectory are visualized in Fig. 4.

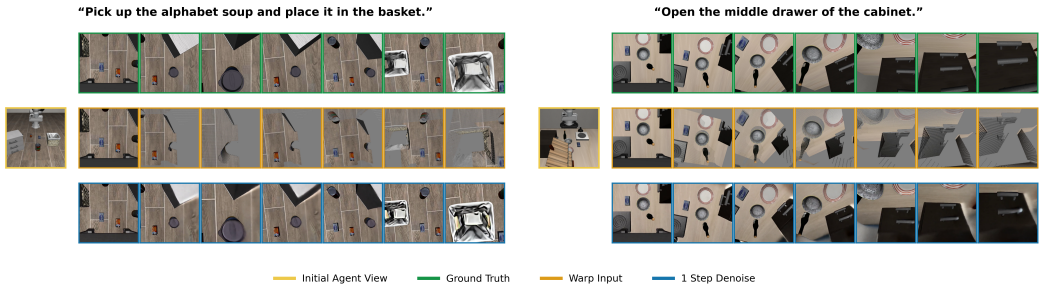

Figure 4: Virtual-view predictions along the planned trajectory. Warped inputs (orange), predictions (blue), and ground truth (green) show progressively improved visibility; RGB decodes are diagnostic only.

The initial agent view is often occluded and does not clearly expose the task-relevant object or interaction region. As the predicted trajectory proceeds, the virtual camera progressively shifts toward more informative regions, revealing the target object, receptacle, or manipulation interface from more favorable perspectives. For instance, the virtual trajectory gradually exposes both the alphabet soup and the basket in the placement task, and moves toward the cabinet to reveal the middle drawer in the drawer-opening task.

The warped inputs provide a coarse geometric projection under the predicted viewpoints, while a single denoising step substantially restores appearance and scene structure necessary for manipulation. The average hole rate of LIBERO-Occ is 39.29%, with peak values reaching 74.47%. Despite these large holes, the denoised predictions still recover the key object layouts and interaction regions, indicating that the trajectory predictor successfully identifies where to observe and that the world model can translate these virtual viewpoints into task-relevant visual evidence for downstream action generation.

5 Conclusion

Rather than relying only on the currently visible camera frames, AWAM predicts a task-conditioned virtual camera trajectory and uses a geometry-conditioned world model to imagine observations along this trajectory. The resulting representations are compressed into a persistent world memory and jointly used with live observations during closed-loop action prediction. Experiments across occlusion-focused and distribution-shift benchmarks show that this additional predictive context improves manipulation robustness when task-relevant objects or interaction regions are poorly observed. These results suggest that world models can serve not only as predictors of future dynamics, but also as internal perceptual mechanisms for querying spatial information that is missing from the current observation. Future work could selectively refresh memory when large state changes are detected, extending virtual active perception to longer-horizon and more dynamic manipulation.

AI USE STATEMENT

Generative AI tools were used to assist with programming, including code drafting, debugging, and implementation support, and to improve manuscript clarity, grammar, and presentation. They were not used to autonomously generate experimental results or determine the scientific conclusions of this work. All AI-assisted code was reviewed, tested, and validated by the authors before use in experiments, and AI-assisted manuscript edits were manually reviewed for technical accuracy, consistency with the underlying experiments, and appropriate attribution. The authors take full responsibility for the final content, including text, claims, code, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, sensitive personal data, or applications with foreseeable societal risks. Experiments are conducted on publicly available datasets and widely used pretrained models under their respective licenses.

REPRODUCIBILITY STATEMENT

Multiple steps are taken to support reproducibility. The model architectures, training data and objectives, and evaluation protocols are specified in the paper, with additional implementation details provided in the appendix. The code will be released upon acceptance.

REFERENCES

Yongjie Bai, Zhouxia Wang, Yang Liu, Kaijun Luo, Yifan Wen, Mingtong Dai, Weixing Chen, Ziliang Chen, Lingbo Liu, Guanbin Li, and Liang Lin. Learning to see and act: Task-aware virtual view exploration for robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13386–13396, 2026.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35101–35113, June 2026.

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*, 2025.

Anthony Brohan et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

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. *The International Journal of Robotics Research*, 2025.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, June 2026.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan

Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.090.

Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt-2: Learning precise manipulation from few demonstrations. *arXiv preprint arXiv:2406.08545*, 2024.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 2025. URL https://proceedings.mlr.press/v267/hu25g.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Taishan Li, Jiwen Zhang, Siyuan Wang, Xuanjing Huang, and Zhongyu Wei. Libero-occ: Evaluating and improving vision-language-action models under scene-induced occlusion via viewpoint imagination. *arXiv preprint arXiv:2606.10862*, 2026.

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*, pp. 8141–8151, 2026.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Tao Xu, Runhao Zhang, Zhijian Huang, Jiayi Guan, Jiaxin Wang, Yifan Ding, Yong-Lu Li, Long Chen, Guang Chen, and Jinghui Lu. UniviewVLA: A unified multiview vision-language-action model with world modeling. *arXiv preprint arXiv:2606.21501*, 2026. URL https://arxiv.org/abs/2606.21501.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Mengchen Zhang, Tong Wu, Jing Tan, Ziwei Liu, Gordon Wetzstein, and Dahua Lin. Gendop: Auto-regressive camera trajectory generation as a director of photography. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18229–18239, 2025a.

Xinyu Zhang, Yuhan Liu, Haonan Chang, Liam Schramm, and Abdeslam Boularias. Autoregressive action sequence learning for robotic manipulation. *IEEE Robotics and Automation Letters*, 10(5): 4898–4905, 2025b. doi: 10.1109/LRA.2025.3550849.

Yisu Zhang, Chenjie Cao, Tengfei Wang, Xuhui Zuo, Junta Wu, Jianke Zhu, and Chunchao Guo. WorldStereo: Bridging camera-guided video generation and scene reconstruction via 3D geometric memories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 40327–40339, 2026.

A Appendix

A.1 Additional Implementation Details

A.1.1 Camera Representation and Virtual Trajectory

Coordinate convention. The static camera and wrist camera are denoted by s and w, respectively. A camera pose $T_i^w = [R_i \mid p_i] \in \mathrm{SE}(3)$ maps camera coordinates to the world frame. All poses used by the geometry pipeline follow the OpenCV camera convention, with x pointing right, y down, and z forward. Metric depth and camera translation use the same physical unit.

As described in 3.2, the trajectory predictor operates on poses relative to the initial wrist-camera pose:

$$\overline{T}_i = (T_0^w)^{-1} T_i^w, \qquad T_i^w = T_0^w \overline{T}_i. \tag{9}$$

This representation removes dependence on the absolute world-frame origin. The world-model renderer consumes the corresponding world-to-camera extrinsics obtained by rigid inversion.

Trajectory representation. Demonstrated wrist-camera trajectories are first temporally resampled. Each relative pose is represented by rotation, normalized translation, camera-related auxiliary coordinates, and a trajectory-scale variable. The normalized coordinates are discretized into trajectory tokens and modeled autoregressively.

For a normalized scalar $x \in [-1, 1]$, the reference implementation uses

$$q(x) = \left\lfloor \mathrm{clip}\left(\frac{B(x+1)}{2}, 0, B\right) \right\rfloor, \qquad B = 256, \tag{10}$$

which gives integer values from 0 to 256. The decoded trajectory is converted back to metric scale and anchored to the observed initial wrist-camera pose using Eq. 9.

Trajectory predictor. The trajectory generator is initialized from a text–RGB-D camera-trajectory model. The static-camera RGB image, metric depth map, and language instruction are independently encoded and projected into a shared conditioning sequence. A causal transformer autoregressively predicts the trajectory tokens using the teacher-forced objective of the main paper.

The visual and language backbone encoders remain frozen during trajectory adaptation, while the trajectory-specific projection layers and causal transformer are optimized for the manipulation-domain trajectories.

Selecting virtual viewpoints. Rather than sending every predicted pose to the world model, $K = 21$ representative poses are retained along the trajectory. To avoid allocating most of the view budget to nearly stationary trajectory segments, viewpoints are sampled according to accumulated camera motion.

For two consecutive poses, the translation and rotation increments are computed as

$$d_i = \|p_i - p_{i-1}\|_2, \tag{11}$$

$$\vartheta_i = \arccos\left[\mathrm{clip}\left(\frac{\mathrm{tr}(R_{i-1}^{\mathsf{T}} R_i) - 1}{2}, -1, 1\right)\right]. \tag{12}$$

Translation and rotation increments are independently normalized by their trajectory totals before being combined. The selected poses are approximately uniformly distributed over accumulated camera motion and include the trajectory endpoints.

Importantly, these poses are *virtual queries*. The robot never physically executes the predicted camera trajectory before manipulation.

A.1.2 Geometry-Conditioned Predictive Memory

Initial geometric evidence. At episode initialization, the calibrated static- and wrist-camera RGB-D observations are back-projected into a joint colored point cloud. For camera $v \in \{s, w\}$ and pixel

$\widetilde{u} = (u_x, u_y, 1)^\mathsf{T}$ with valid depth $D_0^v(u)$,

$$X_0^v(u) = R_0^v \left(D_0^v(u)(K^v)^{-1}\widetilde{u} \right) + p_0^v. \tag{13}$$

The two point sets are merged into the initial geometry $\mathcal{P}_0$.

Rendering $\mathcal{P}_0$ from a predicted viewpoint preserves appearance for surfaces supported by the initial observations, but leaves holes where the initial cameras contain no geometric evidence. This distinction motivates the predictive world model: point-cloud rendering can rearrange observed surfaces, but cannot reconstruct surfaces that were never observed.

Geometry-conditioned world model. WorldStereo (Zhang et al., 2026) is used as the predictive visual backbone and adapted to manipulation scenes. The world model receives the coarse geometric rendering, missing-region mask, camera information, and reference observations, and predicts view-consistent representations along the virtual camera trajectory. The generation targets are static multi-view observations of the episode-initial scene. Supervision is obtained by holding the episode-initial simulator state fixed and rendering additional calibrated camera poses without advancing the environment dynamics. Thus, differences across target views are caused by camera motion rather than object or robot motion.

The generation target corresponds to alternative observations of the initial scene. The robot and scene objects are treated as stationary while the virtual camera changes pose. Thus, the predicted sequence should be interpreted as viewpoint-conditioned scene imagination rather than an action-induced future rollout.

The policy consumes intermediate world-model features instead of generated RGB frames. RGB decoding is used only for the qualitative visualizations in Fig. 4 of the main paper.

Predictive feature extraction. For each of the $K = 21$ virtual views, intermediate world-model features are spatially pooled to a 16×16 grid. In the reference implementation, this produces

$$Z \in \mathbb{R}^{21 \times 256 \times 5120}. \tag{14}$$

The features remain separated by viewpoint during extraction.

Memory compression. The predictive features from the virtual trajectory contain dense spatial information across multiple viewpoints. Rather than exposing all features directly to the action policy, they are compressed with a lightweight query-based memory encoder.

For each virtual view, a small set of learned queries first aggregates its spatial features through cross-attention. Queries occupying the same slot then exchange information across the virtual-view sequence through inter-view attention.

The outputs from all virtual views are flattened into a compact memory M_0, which is cached for the duration of the episode. This representation preserves trajectory-level predictive context. Detailed architectural dimensions of the reference implementation are reported in Table 6.

A.1.3 Memory-Conditioned Closed-Loop Policy

Live visual encoding. At each replanning step, the latest static- and wrist-camera RGB observations are encoded by a shared visual backbone. Camera-specific embeddings preserve the identity of the two visual streams, after which a language-conditioned resampler converts the spatial features into a fixed-size set of live visual tokens L_t.

Unlike the predictive memory, which is constructed once from the initial observation, L_t is recomputed from fresh observations during execution. It therefore provides the policy with up-to-date information about the evolving manipulation state.

Memory retrieval. Predictive information is retrieved from the cached memory using residual cross-attentionwhere the live representation provides the attention queries and the cached memory provides keys and values.

Although M_0 remains fixed during an episode, the attention weights depend on the current observation. Consequently, different predictive information can be retrieved as the manipulation state evolves. The residual pathway also preserves direct access to the current visual evidence.

Table 6: Reference architectural settings used in the LIBERO implementation of AWAM. These values specify the implementation used in our experiments and are reported for reproducibility.

Component	Setting
Virtual-view queries	21 viewpoints
World feature resolution	16×16 tokens per view
World feature width	5120
Memory feature width	384
Memory encoder	2 layers, 8 attention heads
Compressed memory	294 tokens
Live visual representation	224 tokens, width 384
Memory retrieval	8-head cross-attention
Memory lifetime	One episode
LIBERO action horizon	8 actions

A.2 Reference Architecture Settings

Table 6 summarizes the architectural dimensions of the reference LIBERO implementation. These values are reported for reproducibility and should be interpreted as implementation settings rather than separate methodological contributions.

A.3 Training Procedure

The modules are optimized sequentially rather than through end-to-end backpropagation. The trajectory predictor is trained first, followed by world-model adaptation and then the action-policy stages.

Trajectory model. The trajectory model is initialized from a pretrained text–RGB-D camera trajectory generator and adapted using demonstrated wrist-camera paths. The visual and language backbone encoders remain frozen, while the trajectory-specific projection and autoregressive prediction layers are optimized with the teacher-forced objective described in the main paper.

World model. The geometry-conditioned world model (Zhang et al., 2026) is adapted to manipulation scenes while retaining its pretrained geometric and camera-conditioning capabilities. Training targets correspond to alternative observations of the same initial scene under different camera poses. Thus, the model learns to predict viewpoint-conditioned scene representations rather than robot-induced future states.

During action-policy training, world-model features are precomputed and cached. This avoids repeatedly running the large predictive model during policy optimization and provides a fixed representation interface between predictive perception and action learning.

Action policy. Training is divided into an interface-alignment stage and an action-model adaptation stage. The first stage optimizes the live visual projection, resampler, memory encoder, and memory-fusion modules while keeping the pretrained action denoiser fixed. The second stage freezes these connection modules and adapts the diffusion action model.

A.4 Additional Evaluation Details

A.4.1 LIBERO-Occ

LIBERO-Occ evaluates robustness to scene-induced occlusion across the Spatial, Object, Goal, and Long suites. All official benchmark initial states are evaluated, corresponding to 500 rollouts for each suite and 2,000 rollouts in total. Success is determined by the environment-defined task success criterion.Example rollouts are shown in Fig. 5

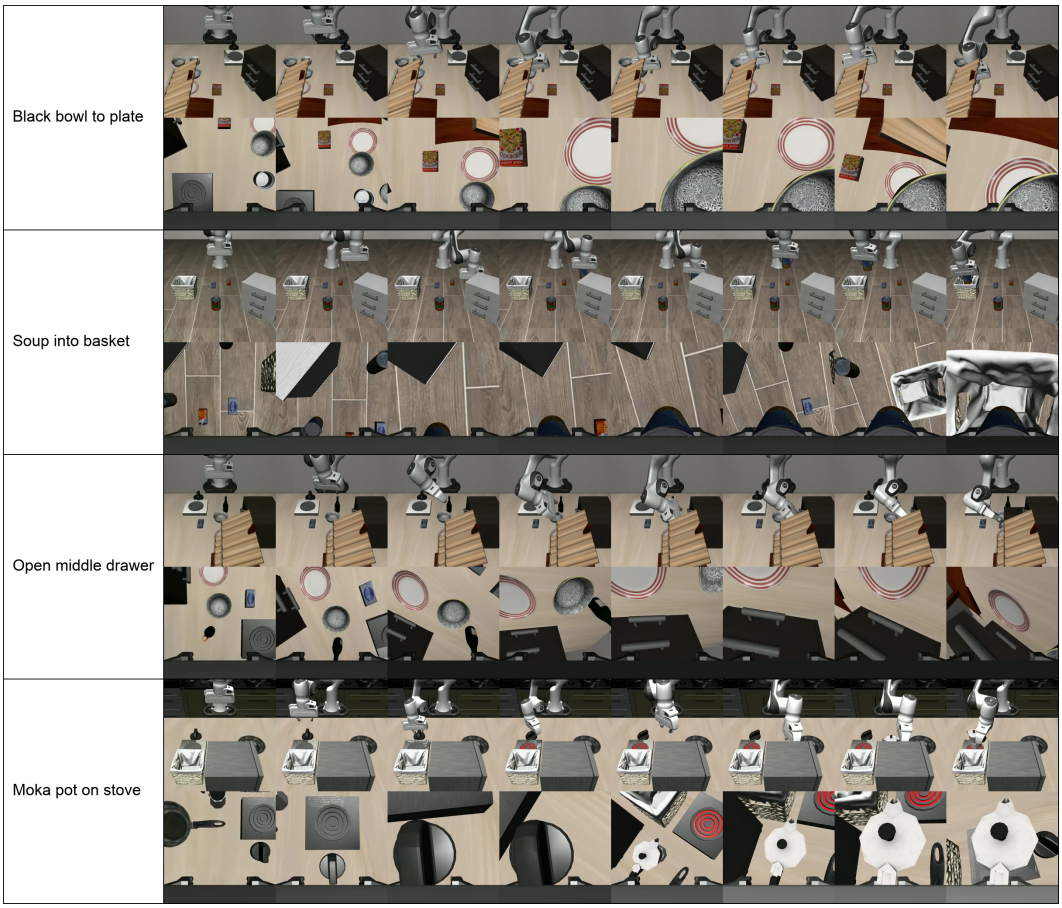

Figure 5: Example rollouts on LIBERO-Occ.

A.4.2 RLBench-OG Occlusion1

Evaluation uses the ten tasks of the RLBench-OG Occlusion1 setting: basketball in hoop, scoop with spatula, take plate off colored dish rack, water plants, block pyramid, solve puzzle, take USB out of computer, close drawer, straighten rope, and toilet seat down. Example rollouts are shown in Fig. 6.

For RLBench-OG Occlusion1, models are trained and evaluated on the occluded task configurations following the benchmark protocol. RLBench-OG uses a control interface different from LIBERO. The same AWAM mechanism—trajectory-conditioned predictive perception, memory construction, and memory-conditioned closed-loop action generation—is retained with the benchmark-specific controller.

A.4.3 Real-World Task Details

Three manipulation tasks are evaluated on an ARCHER-Y6 arm with a GP-100 gripper and scene-level and wrist-mounted Intel RealSense D435 cameras. Each task is evaluated over 30 trials. AWAM constructs its predictive world memory once from the initial observations and retrieves this context alongside live RGB observations and proprioception during closed-loop control. The predicted camera trajectory specifies virtual viewpoints; it is not executed as a separate physical inspection motion. The third-person views in Figure 7 illustrate early and late execution stages. Descriptions of initial target occlusion below refer to the robot-camera observations.

Retrieving an apple from a box. A red apple is placed inside an open box, whose walls partially obscure the apple in the initial robot observations. The challenge is to infer the grasp region



Figure 6: Example rollouts on RLBench-OG Occlusion1.

and its relation to the surrounding walls from incomplete visual evidence. Task-conditioned virtual view planning is designed to provide predictive context for the partially observed box interior, while geometry-conditioned world memory retains these spatial features for action generation. Live observations support gripper positioning and extraction as the scene changes. The selected late frame shows the apple held above the rim, making the retrieval outcome visible from the external viewpoint.

Picking a bottle from clutter and placing it in a box. A plastic bottle stands on a tabletop among a bowl, an apple, and smaller objects, with an open receiving box nearby. The robot should grasp the bottle, lift it from the clutter, transfer it to the box, and release it inside. This task combines target selection in a visually cluttered scene with coordination between a grasp location and a separate placement region. Also, half of the bottle lies outside the initial robot view, thus complicating the interpretation of the bottle's local geometry and the gripper approach. AWAM's task-conditioned virtual-view predictions are intended to complement the initial observations with spatial context, which the world memory retains throughout the transfer. Memory retrieval complements current observations, while closed-loop action prediction follows the changing bottle and gripper states. The late frame shows the bottle standing inside the receiving box as the gripper withdraws.

Grasping an occluded block and placing it in a box. A green block is placed on the tabletop beside a box, which initially occludes the target in the robot-camera observations. The robot approaches the block, closes the gripper around it, lifts it above the box opening, and deposits it inside. The box therefore serves both as an occluder and as the destination, requiring the policy to relate the initially hidden grasp region to the visible container geometry. The cached memory remains available as live observations reveal the target and guide the subsequent grasp and transfer. The selected late frame captures the block held above the opening before release; completion additionally requires depositing the block inside the box.

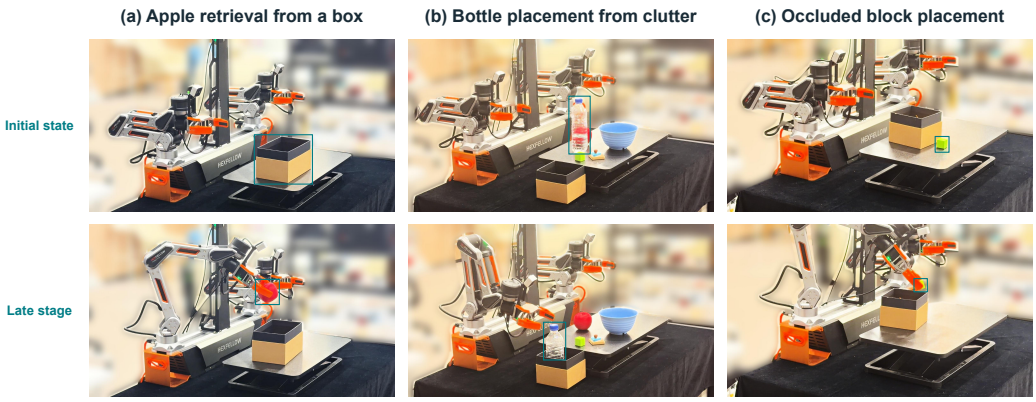

Figure 7: Third-person views (for record only) of real-robot inference on three real-world tasks, with one early and one late frame per task.

A.5 LIMITATIONS

The current system relies on calibrated cameras and depth information to construct geometry-aware virtual observations, which introduces sensing assumptions beyond purely image-based policies. Besides, the current system constructs M_0 once from the initial scene and keeps it fixed throughout manipulation. When objects are substantially displaced or articulated, parts of the cached memory may become stale. Fresh live observations provide the primary control feedback, but the memory itself is not explicitly updated. Dynamically scheduled memory reconstruction is a promising direction for long-horizon tasks.