AWAM: AGENTIC WORLD ACTION MODEL FOR EMBODIED INTELLIGENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing Vision-Language-Action (VLA) models and World Action Models (WAMs) typically rely on end-to-end decision-making architectures without explicit agentic reasoning. Although WAMs provide foresight through future prediction, they lack agentic closed-loop reasoning and remain vulnerable to error accumulation in long-horizon, compositional, and out-of-distribution tasks. To address this limitation, we introduce AWAM, an Agentic World Action Model whose dual-brain architecture comprises an Agentic Reasoning Brain and a World–Action Brain, integrating high-level reasoning, world modeling, and action generation. The Agentic Reasoning Brain reasons over task goals, observations, and retrieved experience to generate and refine structured intents. Conditioned on these intents, the World–Action Brain jointly predicts future world states and executable actions, distilling the predicted physical consequences of these actions into world evidence that is fed back to the Agentic Reasoning Brain. This interaction establishes a unified closed loop in which reasoning and world modeling constrain and refine each other to guide action generation. During training, AWAM generates multiple candidate intents conditioned on the same physical state and evaluates them through simulation rollouts. At inference time, AWAM adopts a single-intent execution strategy and re-deliberates when necessary. Experiments across multiple benchmarks demonstrate AWAM's superior manipulation performance, with strong capabilities in long-horizon execution, compositional generalization, out-of-distribution robustness, and failure recovery.

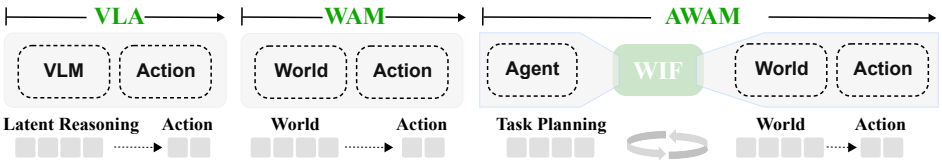


Figure 1: Comparison of VLA, WAM, and AWAM architectures. AWAM couples agentic reasoning with world action brain via a dual-brain closed loop. WIF denotes World Imagination and Feedback.

1 INTRODUCTION

Recent Vision–Language–Action models (VLAs), such as RT-2 and OpenVLA, use vision–language pretraining and diverse robot data to support general-purpose embodied control Brohan et al. (2023); Kim et al. (2024). However, most VLAs map observations and instructions directly to actions without explicitly modeling action consequences or maintaining persistent task states. This limits their ability to handle long-horizon dependencies, skill composition, and environmental perturbations. In such settings, accumulated errors can lead to task-stage confusion and task failure.

World Action Models (WAMs) extend this paradigm by jointly modeling future world states and robot actions, thereby learning physical priors that support action generation Ye et al. (2026b); Wang et al. (2026a). These learned representations can improve action prediction even without explicit video generation at inference time Yuan et al. (2026). However, world modeling provides foresight but not agency. Predicting action consequences alone does not inherently enable WAMs to identify futures worth exploring, compare strategies, or revise decisions based on execution feedback. World

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.

modeling alone therefore does not provide the agentic closed-loop reasoning needed to maintain global objectives, resolve cross-stage dependencies, and recover from failures.

Recent approaches augment frozen VLAs or WAMs with large language model agents for task decomposition, skill orchestration, and experience retrieval Zhang et al. (2026); Gu et al. (2026). However, these systems largely rely on external orchestration, coupling agents and policies through textual subgoals or primitive invocations while providing only coarse execution feedback to the agents. Such interfaces limit the bidirectional exchange of high-level intents and fine-grained physical feedback. Moreover, interaction experience is primarily stored in external memory rather than internalized by the underlying model to improve its physical capabilities.

This progression highlights a missing closed loop among reasoning, world modeling, and action: reasoning should identify futures worth imagining, while world modeling should assess their physical feasibility. To this end, we introduce AWAM, a dual-brain architecture comprising an Agentic Reasoning Brain and a World–Action Brain. The former generates structured physical intents from task goals, observations, and retrieved experience. Conditioned on these intents, the latter jointly predicts future world states and executable actions and feeds back physical evidence to refine high-level decisions. This bidirectional interaction unifies reasoning, world modeling, and action generation within a coherent closed loop. The model's evolution is illustrated in Fig 1.

We further develop a lightweight simulator-grounded preference post-training method. Multiple candidate intents conditioned on the same physical state are evaluated through simulation rollouts, yielding preference supervision based on task outcome, interaction state, and physical risk. We first train the dual-brain architecture with imitation learning and physical prediction objectives. During online adaptation, multiple candidate rollouts are evaluated by the Physical Head, which predicts task outcome, interaction state, and physical risk. With gradients detached from the world imagination pathway, these predictions are converted into structured world evidence for Agent Memory updates and serve as rewards for FPO++(Yi et al., 2026) optimization of the World–Action Brain. At inference time, AWAM executes a single intent and selectively re-deliberates based on execution feedback, avoiding repeated multi-candidate search. Experiments across multiple benchmarks demonstrate consistent improvements over existing approaches, highlighting the effectiveness of AWAM. Our main contributions are summarized as follows:

- We introduce AWAM, a dual-brain architecture for embodied decision making that unifies high-level reasoning, predictive world modeling, and embodied action generation through bidirectional interaction between an Agentic Reasoning Brain and a World–Action Brain.
- We develop a bidirectional interface that connects structured physical intents with world evidence, enabling the Agentic Reasoning Brain to condition the World–Action Brain on task-relevant intents and refine decisions using predicted physical consequences.
- We introduce an agentic data construction pipeline that converts robot demonstrations into reasoning-guided manipulation experiences via a Task–Skill–Primitive hierarchy, primitive-conditioned supervision, and physical feedback, enabling joint learning of reasoning, world modeling, and embodied action generation.
- We validate AWAM across multiple robotic manipulation benchmarks and systematically evaluate its long-horizon execution, compositional generalization, out-of-distribution robustness, failure recovery, and inference efficiency.

2 Related Work

2.1 Vision–Language–Action Models

Vision–Language–Action models have progressed from multi-task policies toward generalist control through scalable robot learning and the transfer of Internet-scale vision–language knowledge Brohan et al. (2022; 2023). Cross-embodiment data aggregation, large-scale pretraining, and efficient adaptation have further broadened their applicability across robots and tasks Open X-Embodiment Collaboration (2023); Octo Model Team et al. (2024); Kim et al. (2024). More recently, flow-based generative modeling of continuous action chunks has enabled fine-grained control and modeling of multimodal action distributions Black et al. (2024). Despite these advances in data scaling, knowledge transfer, and action modeling, VLAs primarily map observations and instructions directly to

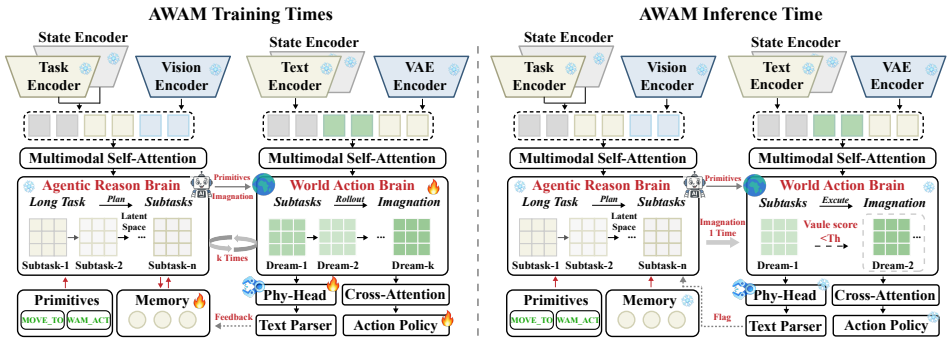


Figure 2: Overview of AWAM. Left: Training evaluates multiple intents via world imagination to update the World–Action model and memory. Right: Inference retrieves experience to generate a single intent by default, adaptively refining it with world evidence and execution feedback.

actions, with limited capacity to explicitly model future physical consequences, maintain persistent task states, or perform closed-loop replanning.

2.2 World Action Models

World Action Models extend direct action prediction by jointly modeling future world states and robot actions. Explicit video prediction captures action-conditioned environmental dynamics, while large-scale video-generation priors help models learn transferable physical dynamics Ye et al. (2026b). Recent developments increasingly emphasize control-oriented world representations through action-relevant future modeling and latent-space transition prediction Ye et al. (2026a); Luo et al. (2026). Integrating predictive knowledge during training can further reduce reliance on costly explicit video generation at inference time Yuan et al. (2026). These advances improve physical foresight and data utilization. Nevertheless, existing WAMs remain centered on finite-horizon world–action prediction, with limited support for agentic reasoning, strategy deliberation, and feedback-driven decision revision to achieve global objectives.

2.3 Agent-Augmented World Action Models

Integrating agents with robotic policies initially relied on language models to compose predefined skills, executable programs, or spatial value fields, linking semantic planning with low-level control Ahn et al. (2022); Liang et al. (2023); Huang et al. (2023). As local manipulation capabilities improved, this integration expanded to task decomposition, experience memory, execution monitoring, and failure recovery, with task-state maintenance and world evidence supporting long-horizon decisions Zhang et al. (2026); Gu et al. (2026). However, these approaches largely rely on system-level orchestration through task graphs, textual subgoals, and execution feedback, leaving reasoning and world–action modeling loosely coupled. AWAM instead conditions world–action modeling on structured physical intents and returns predicted physical consequences to the Agentic Reasoning Brain, enabling bidirectional dual-brain collaboration.

3 Method

3.1 Problem Formulation

Interaction Data. We define the decision context as $x_t = (g, o_t, h_t, m_t)$, where g denotes the task instruction, o_t the current observation and robot state, h_t the interaction history, and m_t the experience retrieved from memory $\mathcal{M}_t$. Given x_t, the frozen Agentic Reasoning Brain proposes K structured intents $\{z_t^k\}_{k=1}^K$. The World–Action Brain predicts their physical consequences and generates corresponding actions. Each candidate yield a execution trajectory τ_t^k and a success label $r_t^k \in \{0, 1\}$. Collected across tasks and decision steps, these records form $\mathcal{D} = \{(x_t, z_t^k, \tau_t^k, r_t^k)\}$.

Learning Objective. We aim to improve full-task success by adapting the parameters θ of the World–Action Brain π_W and updating experience memory while keeping the Agentic Reasoning

Brain π_A frozen. Let $\mathbf{Z} = (z_0, \ldots, z_{L-1})$ denote the intents executed over L decision steps and $\mathcal{T}$ the full trajectory composed of segments τ_t. We express the probability to maximize as follows:

$$p_{\theta,\mathcal{M}_0}(R = 1 \mid g) = \int p(R = 1 \mid g, \mathcal{T}) \prod_{t=0}^{L-1} \pi_A(z_t \mid x_t) P_\theta(\tau_t \mid x_t, z_t)\, d\mathbf{Z}\, d\mathcal{T} \tag{1}$$

Here, $\mathcal{M}_0$ is the initial memory, and P_θ is the trajectory distribution induced by π_W and the environment. The factorization assumes that x_t summarizes relevant history and memory, with rollout evidence and memory-update variables jointly marginalized. Adapting π_W aims to improve intent-conditioned execution, while updated memory guides subsequent intents through m_t with π_A frozen. Local labels r_t^k supervise training, whereas R indicates full-task success.

Discussion. Existing approaches connect frozen language-model agents with pretrained VLAs or WAMs through subgoal interfaces and execution feedback Zhang et al. (2026); Gu et al. (2026). The central challenge is to connect high-level reasoning with fine-grained physical evidence and translate interaction experience into improved execution performance. Successful rollouts reveal effective behaviors, while failed rollouts expose limitations in proposed intents and their execution. This motivates our central question: *Can predicted physical consequences and accumulated experience iteratively refine high-level reasoning and improve long-horizon task success while preserving efficient action generation from world representations?*

3.2 Recursive Reason–World Interaction

AWAM couples high-level reasoning with predictive world modeling to generate executable actions, as shown in Fig 2,. At step t, the task context ξ_t remains fixed over J deliberation rounds. In round j, the Agentic Reasoning Brain proposes a primitive intent $\mathbf{z}_t^j \in \mathcal{Z}$ specifying futures to explore. Conditioned on the intent, the World–Action Brain imagines future states in latent space, producing world representations $\mathbf{w}_{t:t+H}^j \in \mathcal{W}$ and structured physical evidence $\mathbf{e}_t^j$ from the Physical Head, including task outcomes, interaction states, and physical risks, to guide the next intent $\mathbf{z}_t^{j+1}$. After J rounds, the Action Policy decodes actions from the world representation $\bar{\mathbf{w}}_t^J$. The World Brain thus links intent to action through physical world, refining decisions before execution.

3.2.1 Agentic Reasoning Brain

The Agentic Reasoning Brain uses a frozen Qwen3.5 (Qwen Team, 2026) to track long-horizon task progress, decompose remaining objectives into ordered subtasks, and select a primitive for the active subtask. Its reasoning is conditioned on the task context and physical evidence from the World Brain.

At step t and round j, we define the reasoning context as $\tilde{x}_t^j = (x_t, \mathbf{e}_t^{j-1})$, combining the decision context with world evidence from the previous round. Let $S = (s_1, \ldots, s_n)$ denote the remaining subtask sequence, with s_1 active, and let $c = (p, \eta)$ denote an invocation of primitive $p \in \mathcal{P}$ with arguments η. The frozen Agentic Reasoning Brain π_A factorizes its reasoning distribution as follows:

$$\pi_A(S, c \mid \tilde{x}_t^j) = \left[\prod_{\ell=1}^{n} \pi_A(s_\ell \mid \tilde{x}_t^j, s_{<\ell})\right] \pi_A(c \mid \tilde{x}_t^j, S) \tag{2}$$

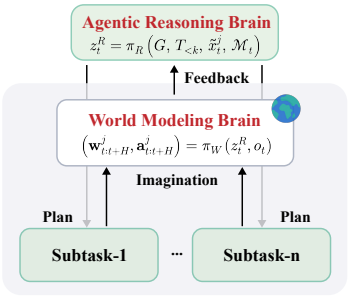


Figure 3: The recurrent reason–world deliberation across subtasks.

The first factor generates dependent subtasks autoregressively, while the second selects the active primitive and its arguments given the plan. Eq 2 groups generated tokens by subtask, including termination in the final group. World evidence $\mathbf{e}_t^{j-1}$ guides revisions to the remaining plan and current primitive invocation.

The primitive library $\mathcal{P}$ includes analytic primitives for free-space motion and robot-state adjustment and a learned `wam_act` primitive for contact-rich interactions. The Agentic Reasoning Brain decomposes the task into primitive-level intents $\mathbf{z}_t^j$, where each primitive invocation c serves as the interface for guiding World–Action Brain imagination and execution. Conditioned on this intent, the World-Action Brain predicts physical evolution and assesses its feasibility.

3.2.2 World Action Brain

The World-Action Brain predicts how the current environment may evolve conditioned on an intent proposed by the Agentic Reasoning Brain. Given the current observation o_t and primitive-level intent $\mathbf{z}_t^j$, it predicts a sequence of latent world states over a horizon of H steps. Rather than reconstructing visually futures, these representations capture decision-relevant physical dynamics, including including task outcomes, interaction states, physical risks, and environmental constraints.

At step t and round j, the it takes o_t and $\mathbf{z}_t^j$ as input and predicts a latent world-state sequence $\mathbf{w}_{t+1:t+H}^j$, from which it extracts physical evidence $\mathbf{e}_t^j$. The joint distribution is as follows:

$$p_{\mathrm{W}}(W, \mathbf{e} \mid o, z) = p_\phi(W \mid o, z)\, p_\psi(\mathbf{e} \mid W) = \left[\prod_{h=1}^{H} p_\phi(w_h \mid o, z, w_{<h})\right] p_\psi(\mathbf{e} \mid W) \tag{3}$$

Here, $p_\phi(W \mid o, z)$ models intent-conditioned physical evolution, while $p_\psi(\mathbf{e} \mid W)$ models physical evidence including task outcomes, interaction states, and physical risks. The dependency $z \to W \to \mathbf{e}$ ensures that intents are evaluated through predicted physical consequences rather than semantic reasoning alone. Aggregated world states guide action denoising without directly passing z to the Action Policy, while world evidence $\mathbf{e}$ refines the Agent's plan and primitive invocation. As illustrated in Fig. 4, multiple rounds of iteration progressively reduce the distribution gap between imagined futures and physical rollouts.

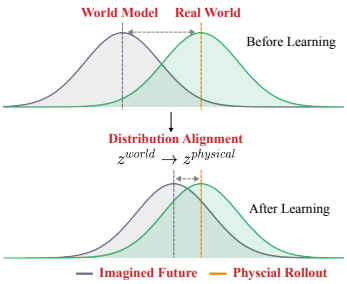


Figure 4: World–physical alignment before and after physical grounding.

3.2.3 Latent-Space Reason–World Recurrence

AWAM couples semantic intent reasoning with physical prediction through recurrent evidence-guided refinement. At round j, let $\tilde{x}_t^j$ denote the reasoning context, including the task context, retrieved experience, and world evidence from the preceding round. Let $\mathcal{C}_t^j = \{z_t^{j,k}\}_{k=1}^K \subset \mathcal{Z}$ denote the candidate intent set. The induced candidate prior $\pi_A(z \mid \tilde{x}_t^j)$ favors intents consistent with the task and interaction history, but does not guarantee physical feasibility.

For each candidate z, the World-Action Brain predicts future world states and extracts physical evidence in terms of task outcomes, interaction states, and physical risks. A physical head estimates $p_W(y = 1 \mid o_t, z)$, where $y = 1$ denotes successful intent execution. Combining this estimate with the semantic prior defines an idealized reason–world posterior:

$$\pi_{\mathrm{RW}}(z \mid \tilde{x}_t^j) := p(z \mid \tilde{x}_t^j, y = 1) = \frac{\pi_A(z \mid \tilde{x}_t^j)\, p_{\mathrm{W}}(y = 1 \mid o_t, z)}{\sum_{z' \in \mathcal{C}_t^j} \pi_A(z' \mid \tilde{x}_t^j)\, p_{\mathrm{W}}(y = 1 \mid o_t, z')} \tag{4}$$

Here, $z \in \mathcal{C}_t^j$ denotes a candidate primitive intent, and π_A represents the normalized prior weights from the Agentic Reasoning Brain. Eq 4 provides a probabilistic reference for balancing semantic preference and physical evidence, increasing the weight of candidates with higher execution reliability. AWAM stores world evidence and value estimates in memory to guide the frozen Agent's future planning and primitive selection. It is shown as in Fig 3.

Proposition 1 (Variational Optimality of the Reference Reason–World Posterior). *Let π_A^j denote the candidate prior on $\mathcal{C}_t^j$, and assume $\ell_W^j(z) := p_W(y = 1 \mid o_t, z) > 0$ on its support. Then the reference posterior in Eq. 4 is the unique optimizer of the following objective:*

$$\pi_{\mathrm{RW}}^j = \underset{q \in \Delta(\mathcal{C}_t^j),\, q \ll \pi_A^j}{\arg\max} \Big[\mathbb{E}_{z \sim q}\big[\log \ell_W^j(z)\big] - D_{\mathrm{KL}}(q \| \pi_A^j)\Big] \tag{5}$$

Here, $q \ll \pi_A^j$ restricts q to the prior's support. The first term favors intents with higher predicted success probabilities, while the second penalizes deviations from the semantic prior. This result characterizes the idealized reference posterior, not the optimality of the implemented memory updates. A proof is provided in the Appendix.

The interaction becomes recurrent as world evidence and physical evaluations are stored in memory to update experience and inform the next reasoning context $\tilde{x}_t^{j+1}$. The subsequent intent distribution is $\pi_A^{j+1}(z) := \pi_A(z \mid \tilde{x}_t^{j+1})$. The frozen Agent uses feedback to refine primitive invocations or retain successful intents. After deliberation, the final world representation guides the Action Policy, while world evidence and retrieved experience refine reasoning and action generation.

3.3 World-Predictive Action Diffusion

AWAM transforms imagined physical futures into executable actions without redundant world prediction. Since the imagined future remains fixed during action generation, its representation is encoded once and reused throughout denoising. The Physical Head evaluates imagined outcomes through task outcomes, interaction states, and physical risks, providing world evidence for selective deliberation and intent refinement.

3.3.1 Diffusion Policy over Predictive World Latents

Let W_t denote the latent world-state sequence associated with the selected intent. We aggregate its multi-level representations into a future context $\mathbf{f}_t = \mathcal{F}_\omega(W_t)$, summarizing predicted physical evolution relevant to action generation. Conditioned on the robot state q_t and future context $\mathbf{f}_t$, the Action Policy generates an action chunk $A_t = (a_t, \ldots, a_{t+C-1})$ via reverse diffusion:

$$p_{\theta_{\mathrm{act}}}(A_t^0 \mid q_t, \mathbf{f}_t) = \int p(A_t^D) \prod_{d=1}^{D} p_{\theta_{\mathrm{act}}}(A_t^{d-1} \mid A_t^d, q_t, \mathbf{f}_t)\, \mathrm{d}A_t^{1:D}, \quad A_t^D \sim \mathcal{N}(0, I), A_t^0 = A_t \quad (6)$$

Here, D is the number of denoising steps, C is the action-chunk length, and θ_{act} denotes the action-denoising parameters. The reverse process progressively transforms the initial noise A_t^D into executable actions A_t^0, reusing the same future context $\mathbf{f}_t$ at every step.

Reusable future context. During action denoising, the imagined future W_t remains fixed while action representations evolve. AWAM computes $\mathbf{f}_t$ once and reuses it at selected interaction layers, with the remaining layers performing action-only refinement. This reduces repeated world computation and dense world–action interaction while retaining future information for action generation.

Action generation from predicted futures. The Action Policy does not directly receive the latent intent z_t. Instead, z_t conditions the future world states predicted by the World-Action Brain, and the resulting future context $\mathbf{f}_t$ is provided to the diffusion policy alongside the robot state q_t.

3.3.2 Physical Evidence-Guided Action Inference

Future-conditioned diffusion generates actions from imagined futures, but high policy likelihood alone does not guarantee successful execution. AWAM therefore evaluates imagined world-action hypotheses through the Physical Head, which predicts structured physical evidence. When additional deliberation is required, AWAM constructs K joint action–future hypotheses $\{(A_t^k, W_t^k)\}_{k=1}^K$ and selects the candidate with the highest execution reliability based on the predicted evidence.

Shared Physical Evaluation. The structured world evidence $\mathbf{e}_t^k$ provides a unified evaluation of imagined futures, including task outcomes, interaction states, and physical risks. These predictions are fed back to the Agentic Reasoning Brain for plan refinement and aggregated into an execution reliability estimate for comparing candidate futures. The same physical evaluation guides both memory-based intent refinement and action selection.

Adaptive inference. AWAM defaults to $K = 1$, avoiding candidate search and enabling efficient future-conditioned action generation. When physical evidence indicates low execution reliability or a mismatch between imagined and observed outcomes, AWAM triggers an additional reason–world deliberation round. The Agentic Reasoning Brain refines its plan and primitive invocation using world evidence, after which the World–Action Brain evaluates the revised intent before the Action Policy generates an updated action chunk. This maintains efficient routine execution while enabling adaptive replanning when needed.

3.4 JOINT LEARNING OF AWAM

AWAM is trained in two stages: supervised fine-tuning followed by online reinforcement learning. Throughout both stages, the Agentic Reasoning Brain remains frozen, while its memory is updated with accumulated world evidence during online adaptation.

Stage I: Supervised Fine-Tuning. During supervised fine-tuning, the World–Action Brain and Physical Head are jointly optimized using demonstration data. Given a target latent world-state sequence W and an action chunk A, we independently sample flow times $\tau_{\mathrm{W}}, \tau_{\mathrm{A}} \sim \mathcal{U}(0,1)$ and Gaussian noises $\epsilon_{\mathrm{W}}, \epsilon_{\mathrm{A}}$. The interpolated states are defined as $W_{\tau_{\mathrm{W}}} = (1-\tau_{\mathrm{W}})W + \tau_{\mathrm{W}}\epsilon_{\mathrm{W}}$ and $A_{\tau_{\mathrm{A}}} = (1-\tau_{\mathrm{A}})A + \tau_{\mathrm{A}}\epsilon_{\mathrm{A}}$. The corresponding target velocities are $u_{\mathrm{W}} = \epsilon_{\mathrm{W}} - W$ and $u_{\mathrm{A}} = \epsilon_{\mathrm{A}} - A$. Let $\widehat{u}_{\mathrm{W}}$ and $\widehat{u}_{\mathrm{A}}$ denote the predicted velocities. The objectives are defined as follows:

$$\mathcal{L}_{\mathrm{WA}} = \lambda_{\mathrm{W}}\mathbb{E}\left[\omega_{\mathrm{W}}(\tau_{\mathrm{W}})\left\|\widehat{u}_{\mathrm{W}} - u_{\mathrm{W}}\right\|_2^2\right] + \lambda_{\mathrm{A}}\mathbb{E}\left[\omega_{\mathrm{A}}(\tau_{\mathrm{A}})\left\|\widehat{u}_{\mathrm{A}} - u_{\mathrm{A}}\right\|_2^2\right] \tag{7}$$

The Physical Head predicts structured physical evidence from imagined world states. Given physical labels $Y_P = \{y_c\}_{c\in\mathcal{C}_P}$, its objective is as follows:

$$\mathcal{L}_{\mathrm{PHY}} = -\sum_{c\in\mathcal{C}_P} \lambda_c y_c \log \hat{y}_c \tag{8}$$

where $\mathcal{C}_P$ denotes the physical prediction categories of task outcomes, interaction states, and physical risks. The supervised stage optimizes the World–Action Brain and Physical Head with $\mathcal{L}_{\mathrm{SFT}} = \mathcal{L}_{\mathrm{WA}} + \lambda_{\mathrm{PHY}}\mathcal{L}_{\mathrm{PHY}}$.

Stage II: Online Reinforcement Learning. After supervised initialization, AWAM performs online adaptation through FPO++ Yi et al. (2026). During this stage, the future prediction component of the World–Action Brain is frozen, while the Action Head and Physical Head are updated from interaction feedback. For each intent, AWAM samples candidate rollouts $\{(A^k, W^k)\}_{k=1}^K$ under the old policy and obtains rewards from execution outcomes and physical evidence. Following the manipulation fine-tuning setting of FPO++, we use per-sample surrogate ratios with PPO clipping:

$$\mathcal{L}_{\mathrm{FPO++}} = -\mathbb{E}\left[\frac{1}{KM}\sum_{k=1}^{K}\sum_{i=1}^{M}\min\Big(\rho_\theta^{k,i}\hat{A}^k, \mathrm{clip}\Big(\rho_\theta^{k,i}, 1-\eta, 1+\eta\Big)\hat{A}^k\Big)\right] \tag{9}$$

where $\hat{A}^k$ is the group-normalized advantage, η is the clipping parameter, and $\rho_\theta^{k,i} = \exp(\ell_{\theta_{\mathrm{old}}}^{k,i} - \ell_\theta^{k,i})$. Here, $\ell_\theta^{k,i}$ is the Stage-I action flow-matching error on (A^k, W^k), summed over the action chunk using the i-th of M flow-time and noise pairs shared by both policies. The Action Head learns from interaction rewards, while the Physical Head retains auxiliary supervision. The online objective is defined as follows:

$$\mathcal{L}_{\mathrm{online}} = \mathcal{L}_{\mathrm{FPO++}} + \lambda'_{\mathrm{PHY}}\mathcal{L}_{\mathrm{PHY}} \tag{10}$$

The two-stage optimization enables AWAM to first acquire general world-action-physical representations from demonstrations and then improve execution reliability through closed-loop interaction.

4 EXPERIMENTS

We systematically evaluate AWAM on public dataset, focusing on complex long-horizon tasks, compositional skill execution, and out-of-distribution generalization. Beyond task success, ablations and process diagnostics assess how dual-brain interaction, world evidence, memory, and value-guided replanning contribute to complex-task execution, failure recovery, and inference efficiency.

Benchmarks. We evaluate AWAM on two complementary benchmarks. RoboDojo assesses generalization, memory, precision, long-horizon execution, and open-vocabulary instruction following in simulation and the real world (Chen et al., 2026). BiCoord tests long-horizon bimanual coordination across subgoals (Peng et al., 2026). Together, they cover diverse task demands, multi-stage execution, and compositional coordination.

Baselines. We compare two categories of methods. VLA policies include ACT Zhao et al. (2023), RDT-1B (Liu et al., 2025), OpenVLA-OFT (Kim et al., 2025), X-VLA Zheng et al. (2026),

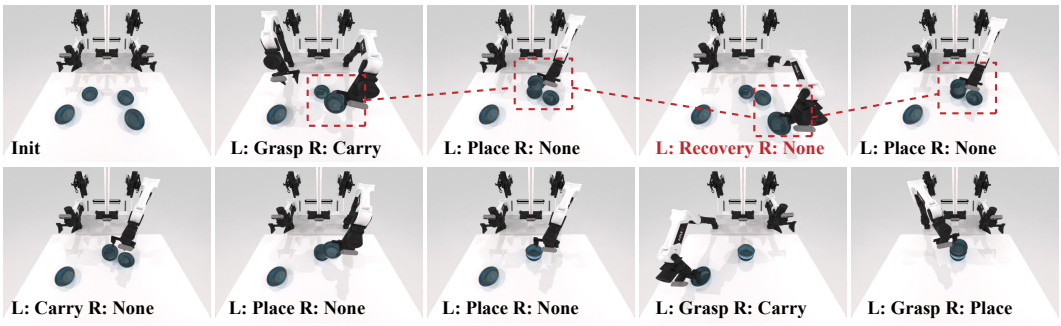


Figure 5: Reasoning and failure recovery in AWAM. The agent coordinates per-arm primitives, recovers from a failed case, and resumes execution. Red annotations highlight the recovery sequence.

Table 1: Performance comparison on RoboDojo across generalization, precision, long-horizon execution, memory, and open-vocabulary instruction following, measured by Score/Sucess Rate.

Method	Policy	Generalization	Precision	Long-Horizon	Memory	Open	Average
ACT	VLA	0.69/0.56	0.85/0.00	1.73/0.92	1.65/0.13	0.00/0.00	0.98/0.32
π_0	VLA	3.94/2.56	3.56/0.75	6.19/2.00	3.47/2.11	0.25/0.25	3.48/1.53
RDT-1B	VLA	0.56/0.33	0.38/0.00	1.13/0.00	0.49/0.33	0.00/0.00	0.51/0.13
X-VLA	VLA	10.48/6.78	18.32/12.00	16.53/9.75	4.76/3.56	0.55/0.50	10.13/6.52
$\pi_{0.5}$	VLA	13.37/8.17	12.40/5.50	23.54/14.67	5.78/4.56	1.98/1.67	11.41/6.91
Spatial Forcing	VLA	14.12/9.33	17.33/10.58	23.26/14.58	5.43/4.11	1.78/1.58	12.38/8.04
Hy-Embodied-0.5	VLA	11.77/8.39	13.81/8.00	25.74/14.92	13.37/12.11	0.65/0.58	13.07/8.80
Fast-WAM	WAM	2.34/1.11	1.96/0.00	9.14/5.17	3.55/3.44	0.42/0.42	3.48/2.03
GigaWorld-Policy	WAM	5.34/2.89	6.15/1.83	15.51/8.92	3.46/2.22	0.54/0.50	6.20/3.27
X-WAM	WAM	7.39/3.33	6.72/1.83	17.47/9.08	6.32/4.67	0.57/0.25	7.69/3.83
OpenWAM	WAM	20.71/14.83	18.45/9.25	34.93/25.33	10.41/9.11	1.41/1.08	17.18/11.92
AWAM (Ours)	Agent+WAM	18.13/ 11.53	14.21/ 6.31	40.74/ 29.92	25.22/ 21.69	2.13/ 1.85	20.08/ 14.26

π_0 (Black et al., 2024), $\pi_{0.5}$ Black et al. (2025), Hy-Embodied-0.5 Team et al. (2026), and Spatial Forcing Li et al. (2026). WAMs for future-informed action prediction include GigaWorld-Policy (Ye et al., 2026a), Fast-WAM (Yuan et al., 2026), X-WAM (Guo et al., 2026), and OpenWAM (Wang et al., 2026b). All baselines follow official configurations.

4.1 Main Experimental Results

RoboDojo. AWAM achieves the best overall performance, with notable gains on memory-dependent and long-horizon tasks. VLA baselines remain competitive on short-horizon precision tasks but struggle with persistent state tracking, multi-stage composition, and failure recovery. WAMs provide physical foresight through future prediction but remain constrained by finite-horizon decision making, while agent-augmented VLAs lack fine-grained physical feedback for revising high-level plans. In contrast, AWAM combines experience retrieval with recurrent reason–world interaction to track task progress across stages and revise unreliable intents based on predicted physical consequences, improving long-horizon execution while preserving fundamental manipulation capabilities.

BiCoord. AWAM also performs best on BiCoord, which tests long-horizon bimanual manipulation requiring precise local actions, sustained inter-arm coordination across subgoals, and dynamic role transitions. Direct action-generation methods are susceptible to coordination drift and compounding errors as the task horizon increases. AWAM instead maintains the global task structure through the Agentic Reasoning Brain and uses the Physical World Brain to assess how candidate intents jointly affect bimanual coordination and environment dynamics before generating actions. This recurrent dual-brain interaction improves both global subtask consistency and local physical coordination, resulting in higher full-task success on tightly coupled bimanual tasks.

Agentic Reasoning and Failure Recovery. As shown in Fig. 5, AWAM coordinates per-arm primitives, invokes recovery after a failed placement, and resumes task execution. This sequence illustrates its ability to revise execution plans based on observed outcomes. Unlike direct action-generation VLAs and WAMs without explicit agentic reasoning, AWAM integrates high-level intent refinement with low-level execution, enabling recovery during task execution.

Table 2: Results of multi-task learning on BiCoord benchmark. SR, SSR, and TL denote Success Rate, Stage-Wise Success Rate, and Task Length, respectively.

Task	RDT			OpenVLA-OFT			π_0			AWAM		
	SR ↑	SSR ↑	TL ↓	SR ↑	SSR ↑	TL ↓	SR ↑	SSR ↑	TL ↓	SR ↑	SSR ↑	TL ↓
Balance Roller	15.0	43.0	149	49.0	74.5	137	68.0	81.5	151	78.0	89.0	140
Build Bridge	47.0	50.2	370	2.0	2.0	312	44.0	47.0	289	55.0	65.0	336
Build Tower With Blocks	0.0	1.0	-	0.0	0.0	-	2.0	5.0	650	8.0	21.0	700
Clean Table	0.0	17.5	-	2.0	25.0	612	6.0	46.2	658	18.0	56.8	756
Collect Pens	15.0	68.8	495	5.0	46.2	795	47.0	81.8	457	60.0	86.0	560
Cook	9.0	31.0	569	10.0	25.0	442	14.0	29.5	504	28.0	42.0	420
Divide Block Tower	2.0	12.8	384	0.0	11.1	-	0.0	10.8	-	8.0	27.0	588
Exchange Mics	23.0	35.0	467	66.0	67.5	418	52.0	59.5	446	63.0	69.0	532
Exchange Pots	92.0	96.0	459	53.0	56.0	442	52.0	60.5	476	62.0	67.0	532
Extract Bottom Block To Top	17.0	25.5	192	53.0	55.5	177	40.0	43.5	208	50.0	53.5	364
Fetch Block With Roller	0.0	49.5	-	44.0	70.5	225	43.0	69.5	250	55.0	71.5	252
Handover Block With Bowls	0.0	20.0	-	0.0	3.0	-	4.0	9.0	188	12.0	16.0	336
Jigsaw	0.0	11.5	-	0.0	25.2	-	6.0	36.5	400	16.0	45.5	476
Match Blocks With Signs	1.0	7.0	1216	5.0	8.7	760	8.0	18.7	638	17.0	41.0	616
Place Plate And Cup	40.0	70.0	320	55.0	79.2	315	25.0	66.2	302	40.0	70.8	308
Put Objects Cabinet	31.0	49.0	526	26.0	49.5	486	6.0	24.0	575	20.0	36.0	644
Stack Bowls	3.0	28.3	597	1.0	14.0	400	12.0	28.0	471	26.0	39.5	448
Sweep Block	10.0	10.0	256	44.0	44.0	234	61.0	61.0	253	72.0	72.0	336
Average	16.9	34.8	461	23.1	36.5	411	27.2	43.2	406	38.2	53.8	491

Table 3: Component ablations on Bicoord. Average success rates across all categories.

Model Variant	SR	SSR	TL
Direct Intent-to-Action	29.0	45.0	420
+ Agentic Reasoning	32.0	46.5	628
+ Experience Memory	34.5	49.0	553
+ Physical Feedback	36.0	51.5	504
+ RL Training	38.2	53.8	491

Table 4: Ablation on the number of imagination rollouts during AWAM training.

Number of Rollouts	SR	SSR	TL
Times k=1	36.0	51.5	448
Times k=2	37.4	52.8	476
Times k=3	38.2	53.8	491
Times k=4	38.3	53.9	504
Times k=5	38.5	54.0	532

4.2 Ablation Study

Key Component Ablation. The ablation results in Table 3 show that task completion performance improves as the components are progressively introduced. Compared with the WAM baseline that directly maps intents to actions, adding agentic reasoning improves performance, suggesting that explicit reasoning helps organize task execution. Incorporating experience memory yields further gains, highlighting the value of reusing past experience to support decision-making. Physical feedback provides additional benefits, suggesting that integrating physical evidence into reasoning improves the feasibility of the resulting decisions. Reinforcement learning further improves performance through interaction. Together, these results support the components' complementary roles in high-level reasoning and low-level execution.

Imagination Rollouts Ablation. Table 4 shows that increasing the number of imagination rollouts during training improves both task and subtask success rates, although the gains begin to saturate beyond three rollouts. This trend suggests that additional imagined trajectories provide useful training signals, but their marginal benefits diminish as the rollout count increases. Meanwhile, trajectory length increases, particularly with five rollouts, indicating that higher success rates do not necessarily imply more efficient execution. Three rollouts balance task success and trajectory length.

5 Conclusion

We introduced AWAM, a dual-brain framework that couples agentic reasoning with world modeling through recurrent interaction. The Agentic Reasoning Brain generates structured intents, while the World–Action Brain predicts their consequences, decodes actions from world representations, and provides physical evidence for intent refinement. With the reasoning backbone frozen, AWAM integrates experience memory with world–action learning and reinforcement learning to support single-path execution with adaptive replanning. Experiments on RoboDojo and BiCoord demonstrate improvements in task completion, long-horizon execution, and bimanual coordination. Ablations support the complementary contributions of agentic reasoning, experience memory, physical feedback, and reinforcement learning. Future work will explore open-world tasks, cross-embodiment transfer, and continual learning from physical experience.

AI USE STATEMENT

In this work, we have not used generative AI tools for any tasks with required disclosure, and all such tasks are not applicable to this work. We used generative AI tools only to polish the writing of parts of the manuscript. The authors reviewed all AI-assisted edits to ensure that they preserve the intended technical meaning and contain no unsupported claims. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

For reproducibility, the main paper and Appendices detail our methodology and implementation. We will release the complete source code and model weights.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. URL https://arxiv.org/abs/2204.01691.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

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 *Proceedings of the 9th Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40, 2025. URL https://proceedings.mlr.press/v305/black25a.html.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. URL https://arxiv.org/abs/2212.06817.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. URL https://arxiv.org/abs/2307.15818.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025. URL https://arxiv.org/abs/2506.18088.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, et al. Robodojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026. URL https://arxiv.org/abs/2607.04434v3.

Zhaopeng Gu, Bingke Zhu, Tianxi Lin, Guibo Zhu, Yingying Chen, Kai Wang, Tingyu Yuan, Chaoyang Zhao, Zhaowen Li, Peng Su, and Jinqiao Wang. HarnessWAM: Bridging prediction and deliberation in world action models. *arXiv preprint arXiv:2608.09516*, 2026. URL https://arxiv.org/abs/2608.09516.

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4d world action modeling from video priors with asynchronous denoising. *arXiv preprint arXiv:2604.26694*, 2026. URL https://arxiv.org/abs/2604.26694.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3d value maps for robotic manipulation with language models. In *Conference on Robot Learning*, pp. 540–562, 2023. URL https://arxiv.org/abs/2307.05973.

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. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Robotics: Science and Systems*, 2025. URL https://arxiv.org/abs/2502.19645.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.12276.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500, 2023. URL https://arxiv.org/abs/2209.07753.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a Diffusion Foundation Model for Bimanual Manipulation. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.07864.

Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-H0.7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026. URL https://arxiv.org/abs/2605.00078.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. URL https://arxiv.org/abs/2405.12213.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. *arXiv preprint arXiv:2310.08864*, 2023. URL https://arxiv.org/abs/2310.08864.

Xingyu Peng, Chen Gao, Liankai Jin, Annan Li, and Si Liu. Bicoord: A bimanual manipulation benchmark towards long-horizon spatial-temporal coordination. *arXiv preprint arXiv:2604.05831*, 2026. URL https://arxiv.org/abs/2604.05831.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

HY Team, Xumin Yu, Zuyan Liu, Ziyi Wang, He Zhang, Yongming Rao, Fangfu Liu, Yani Zhang, Ruowen Zhao, Oran Wang, et al. Hy-embodied-0.5: Embodied foundation models for real-world agents. *arXiv preprint arXiv:2604.07430*, 2026.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan

Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Siyin Wang, Junhao Shi, Zhaoyang Fu, Xinzhe He, Feihong Liu, Chenchen Yang, Yikang Zhou, Zhaoye Fei, Jingjing Gong, Jinlan Fu, Mike Zheng Shou, Xuanjing Huang, Xipeng Qiu, and Yu-Gang Jiang. World action models: The next frontier in embodied ai. *arXiv preprint arXiv:2605.12090*, 2026a. URL https://arxiv.org/abs/2605.12090.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, Yue Chen, Xuemin Chi, Donghao Zhou, Qize Yu, et al. Openwam: An open, modular exploration towards systematic world-action model pretraining. *arXiv preprint arXiv:2609.07398*, 2026b.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. GigaWorld-Policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026a. URL https://arxiv.org/abs/2603.17240.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026b. URL https://arxiv.org/abs/2602.15922.

Brent Yi, Hongsuk Choi, Himanshu Gaurav Singh, Xiaoyu Huang, Takara E. Truong, Carmelo Sferrazza, Yi Ma, Rocky Duan, Pieter Abbeel, Guanya Shi, Karen Liu, and Angjoo Kanazawa. Flow policy gradients for robot control. *arXiv preprint arXiv:2602.02481*, 2026. doi: 10.48550/arXiv.2602.02481. URL https://arxiv.org/abs/2602.02481.

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. URL https://arxiv.org/abs/2603.16666.

Yixian Zhang, Huanming Zhang, Feng Gao, Xiao Li, Zhihao Liu, Chunyang Zhu, Jiaxing Qiu, Yuchen Yan, Jiyuan Liu, Wenhao Tang, Zhengru Fang, Yi Nie, Changxu Wei, Yu Wang, Wenbo Ding, and Chao Yu. Harness VLA: Steering frozen VLAs into reliable manipulation primitives via memory-guided agents. *arXiv preprint arXiv:2607.08448*, 2026. URL https://arxiv.org/abs/2607.08448v1.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

Weiheng Zhao, Haoyi Jiang, Xin Shi, Liu Liu, Fan Huang, Zhizhong Su, Wei Sui, and Xinggang Wang. Faster-WAM: Efficient Inference-Time Future Conditioning for Robust World Action Models. *arXiv preprint arXiv:2608.04404*, 2026. URL https://arxiv.org/abs/2608.04404.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kt51kZH4aG.

A Implementation Details

A.1 Dual-Brain Architecture and Interfaces

AWAM integrates a frozen reasoning backbone with trainable world and action modules through structured intents, physical feedback, and experience memory. Sparse world–action interaction and cached future context support efficient action generation, while value estimates guide action selection and adaptive replanning. Table 5 summarizes the architecture, training configuration, and inference settings.

Table 5: Architecture and implementation settings of AWAM.

Category	Configuration	Setting
Reasoning Brain	Reasoning backbone	Qwen3.5-27B
	Backbone parameters	Frozen
	Experience memory	Updated separately
World and Action Modules	Pretrained backbone	Wan2.2-5B
	Initialized components	Video DiT, text encoder, and VAE
	Multi-camera input	Spatially assembled images
	Visual sequence length	9 frames
	Temporal sampling stride	4
	Action chunk length	32
	Action Transformer depth	30 layers
	Action hidden dimension	1024
World–Action Interaction	Interaction stages	8
	Interaction interval	4 stages
	Feature aggregation	Interval KV-Fusion
	Future-context reuse	Cached throughout each action chunk
	Action conditioning	Future context and robot state
Training	World/action training objective	Flow matching
	Optimizer	AdamW
	Initial learning rate	10^{-4}
	Weight decay	0.01
	Learning-rate schedule	Cosine decay
	Training precision	BF16
	Maximum gradient norm	1.0
Action Inference	Numerical integration steps	10
	Guidance scale	1.0

Agentic Reasoning Brain Details. The Agentic Reasoning Brain π_A uses Qwen3.5 with frozen parameters. It combines the task instruction and current observations with interaction history and retrieved experience to plan the subtask sequence and specify the primitive invocation for the next operation. These outputs form the intent representation supplied to the World Brain. Physical evidence and value estimates are recorded in memory, allowing subsequent decisions to incorporate the predicted and observed consequences of earlier intents. The architecture does not require a particular reasoning backbone. Other compatible large language models, can replace Qwen3.5 using the same input and structured-output interfaces.

World–Modeling Brain Details. The video DiT, text encoder, and VAE in π_W are initialized from Wan2.2-5B (Team Wan et al., 2025). Multi-camera images are combined into a shared spatial layout before VAE encoding. The visual sequence comprises nine frames sampled at a temporal stride of four, paired with a 32-action chunk. The action module uses a 30-layer Transformer with a hidden dimension of 1024. Intent information conditions world modeling, while action generation is conditioned on the resulting future context and the current robot state.

World–Action Interaction Details. World features are injected at eight selected stages of the 30-layer action network, with an interaction interval of four. Interval KV-Fusion (Zhao et al., 2026) aggregates video features over the corresponding depth intervals, while the remaining action layers update only action features. The future context is computed once and cached throughout the generation of each action chunk. World evidence is separately returned to π_A for reasoning refine-

ment, while the Physical Head evaluates imagined outcomes to guide action selection and adaptive replanning.

Optimization and Inference. Both visual and action generation are trained using flow matching. We optimize the trainable components with AdamW using an initial learning rate of 10^{-4} and a weight decay of 0.01. The learning rate follows a cosine decay schedule. Training uses BF16 precision and gradient norm clipping with a maximum norm of 1.0. The reasoning backbone remains frozen, while experience memory is maintained separately. At inference time, actions are sampled using 10 numerical integration steps with a guidance scale of 1.0.

A.2 Data Construction and Details

A.2.1 Data Construction

AWAM's dual-brain architecture requires data that support task-level reasoning, Primitive-level manipulation, and physical feedback modeling. We therefore organize robot demonstrations into a Task–Skill–Primitive hierarchy linked to action sequences, derive Primitive-conditioned action supervision, and obtain physical outcome labels through candidate execution.

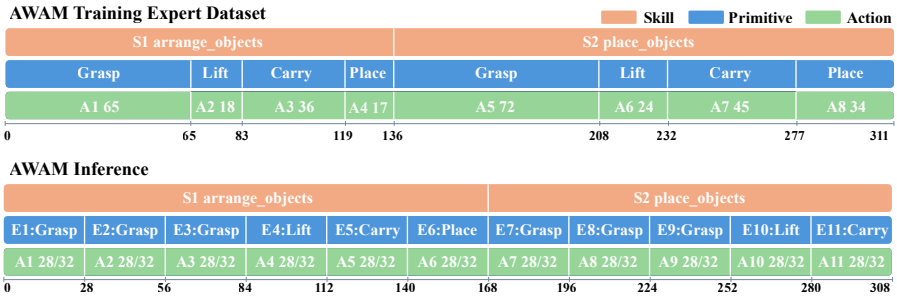


Figure 6: Expert and AWAM execution timelines on the BiCoord bridge task. Rows show Skill, Primitive, and Action segments. AWAM executes 28 steps per 32-step chunk. Timelines are independently normalized, without frame-wise alignment.

Data Sources and Annotation. We use demonstrations from BiCoord and RoboDojo, together with available replay records, and apply the same hierarchical annotation procedure to real-world manipulation trajectories. We first align visual observations, robot states, actions, and physical evidence. Primitive boundaries are identified from changes in motion and gripper state and verified using semantic and physical evidence. For bimanual trajectories, merging left- and right-arm boundaries yields composition units, each containing at most one Primitive per arm and specifying grounded object roles, inter-arm dependencies, and termination conditions. Primitive sequences are then grouped into Skills according to local goals while preserving their task-level execution order, as shown in Fig 6. Shared rules convert each object-grounded Primitive step into a conditioning instruction during both data construction and runtime. Trajectories with incomplete annotations are excluded. As shown in Fig 7, we develop an annotation tool to support hierarchical data annotation, and all annotations are manually verified.

Interactive Action Primitive Annotation Tool. We develop an interactive tool for the structured annotation of action primitives in bimanual manipulation trajectories as shown in Fig 7. The tool displays key events, including gripper state changes and pauses in arm motion, on separate timelines for the left and right arms. Observation frames and key-frame thumbnails help annotators interpret the manipulation process. Annotators can specify skill names, execution order, and dependencies, as well as action primitive sequences for each arm. The annotations can be exported in a structured format to support subsequent data processing and model training.

A.3 Data Formats and Examples

Hierarchical Trajectory Representation. Figure 6 illustrates the resulting Skill–Primitive–Action hierarchy using an expert demonstration and an AWAM execution. Each trajectory associates



Figure 7: Interface of our primitive annotation tool. An interactive event timeline and key-frame viewer support inspection of bimanual trajectories, while the annotation panel records skill names, dependencies, and per-arm primitive sequences.

Skills with Primitive segments and their corresponding action intervals. In the illustrated AWAM execution, each prediction window contains 32 action steps, of which the first 28 are executed before acquiring a new observation for the next prediction. The records retain repeated Primitive invocations and actual execution intervals for feedback analysis and memory construction.

Primitive Action Sample. Figure 8 shows a sample from BiCoord's `balance_roller` task. Both arms are assigned the `Grasp` Primitive with a `coupled` dependency, requiring shared control of the roller. Given the observation at frame 0 and the grounded instruction, the model is supervised using actions from frames 1–32, within the composition interval $[0, 67]$. All 32 target steps are valid, so no padding is required.

Physical Feedback Sample. Figure 9 illustrates a physical feedback record for a model-generated candidate conditioned on the annotated Primitive. It links the initial three-view observations, robot state, and candidate action to execution-derived outcome labels. The target indicates an intermediate state, with no forbidden collision or interaction formation. All five labels are known and included in supervision, as specified by the validity mask.

Memory Sample Figure 10 illustrates a Strategy Memory entry for bowl stacking. It associates a left-arm `Lift` Primitive with the target category `bowl`, the reference category `table`, and its position between `Grasp` and `Carry` within the Skill. The entry captures a strategy pattern from training demonstrations rather than a verified outcome in the current scene. Object categories are bound to specific instances at runtime to support context-dependent planning.

A.4 Experience Memory Details

AWAM maintains external experience memory to provide the frozen Agentic Reasoning Brain with continually updated decision context. As illustrated in Figure 20, this memory is organized into task-specific execution experience and transferable global knowledge. Predicted world evidence, value estimates, and verified execution outcomes enrich these records, enabling the Agent to adapt previous solutions to the current environment.

Memory Organization. Task-specific memory contains execution traces and strategy summaries. Execution traces record subtask ordering, primitive invocations and their arguments, stage transitions, and recovery steps. Strategy summaries capture outcomes, effective decisions, failure causes, and relevant constraints. Global memory stores primitive preconditions, recurring failure patterns,

```
{
  "sample_id": "bicoord__balance_roller__clean__0__w0",
  "benchmark_origin": "bicoord",
  "task": "balance_roller",
  "episode": 0,
  "split": "train",
  "frame_index": 0,
  "composition_interval": [0, 67],
  "wam_condition": "Coordinate both arms to grasp the roller with
    the left arm and grasp the roller with the right arm, maintaining
     their shared control and relative pose.",
  "step": {
    "schema_version": "4.0",
    "dependency": "coupled",
    "left": {
      "primitive": "Grasp",
      "arguments": {
        "controller": "left",
        "target": "the roller"}},
    "right": {
      "primitive": "Grasp",
      "arguments": {
        "controller": "right",
        "target": "the roller"}}},
  "target": {
    "action_ref": {
      "dataset": "/joint_action/vector",
      "dimension": 14,
      "horizon": 32,
      "index_semantics": "native_joint_action[t+1]",
      "start": 1,
      "valid_length": 32},
    "prefix_valid_mask": "32 valid positions"},
  "instruction_variant": "canonical"
}
```


Figure 8: A Primitive-conditioned action sample from BiCoord. A coupled bimanual grasp is supervised by 32 steps of 14-D actions starting at frame 1. Image and proprioceptive references are omitted; task and provenance metadata are excluded from model inputs.

and corrective strategies applicable across tasks. For example, gripper closure without corresponding object motion may prompt re-localization and grasp-pose adjustment. Together, these two memory types support task organization, execution monitoring, and failure recovery.

Sources of Experience. AWAM acquires experience from world prediction and execution in the environment. For each candidate intent, the World Brain provides evidence of feasibility, task progress, and interaction risk, while the value head estimates the associated return. These predictions support intent comparison and plan revision before execution. After execution, updated observations and feedback reveal the actual outcome and any discrepancies from the predicted outcome. Memory distinguishes between predicted information and verified outcomes. Unexecuted candidates are not recorded as successful experience.

Memory Updates. After each interaction, the Agent examines available evidence and execution records to assess task progress, recovery needs, and plan validity. Successful executions provide reusable procedures, while recoverable failures are retained with their corrective steps. Unsuccessful attempts contribute negative experience about conditions and behaviors to avoid. New evidence can be used to revise strategy summaries and improve stored procedures while preserving relevant failure constraints. Raw records support execution tracing, whereas consolidated experience supports subsequent retrieval.

Retrieval and Plan Revision. At decision step t, the Agent retrieves relevant procedures, strategy summaries, and global rules from memory $\mathcal{M}_t$. The retrieved experience m_t is combined with the

```
{
  "candidate_id": "<group-id>/model_same/chunk0",
  "role": "model_same",
  "sample_source": "online_model_execution",
  "condition": "Coordinate both arms to grasp the roller ...",
  "action_valid_rows": 32,
  "model_input": {
    "first_observation_images": {
      "head_camera": "input_head_camera.png",
      "left_camera": "input_left_camera.png",
      "right_camera": "input_right_camera.png"
    },
    "proprio": "proprio.npy",
    "generated_action": "generated.npy"
  },
  "label_order": [
    "success",
    "intermediate",
    "failure",
    "forbidden_collision",
    "interaction_formed"
  ],
  "target": [0, 1, 0, 0, 0],
  "mask": [true, true, true, true, true]
}
```


Figure 9: A physical feedback supervision sample. A model-generated candidate is paired with execution-derived labels. The outcome is intermediate, with no forbidden collision or interaction formation; all five labels are valid. The conditioning instruction is abbreviated.


```
{
  "task": "stack_bowls_two",
  "task_family": "stacking",
  "skill_instruction": "place the selected bowl on the reference
    bowl",
  "skill_relations": ["placed_at"],
  "dependency": "none",
  "primitive_pattern": {
    "left": {
      "primitive": "Lift",
      "target_category": "bowl",
      "reference_category": "table"
    },
    "sequence_context": {
      "previous_primitives": ["Grasp"],
      "next_primitives": ["Carry"],
      "skill_position": 1,
      "skill_length": 4
    }
  },
  "source_split": "train"
}
```


Figure 10: An illustrative Strategy Memory entry. A left-arm `Lift` Primitive is associated with object categories and its `Grasp–Lift–Carry` sequence context within a bowl-stacking Skill. Object categories are bound to scene-specific instances at runtime.

task instruction, current observation, and interaction history to form $x_t = (g, o_t, h_t, m_t)$. Here, h_t records the ongoing execution history, while m_t contains selected experience retrieved from memory. This context guides subtask revision, primitive selection, and recovery planning. Stored

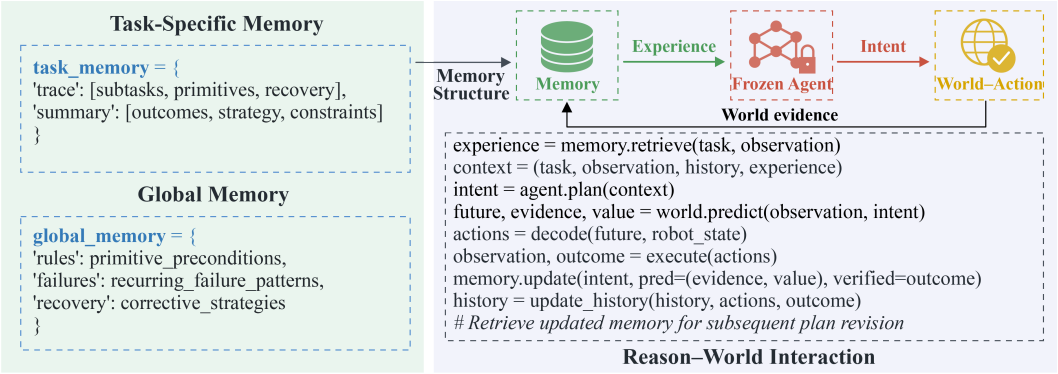


Figure 11: Experience memory in AWAM. Task-specific and global knowledge are updated with world evidence and execution outcomes to guide the frozen Agentic Reasoning Brain.

locations and target poses remain specific to their original scenes, so execution arguments are recomputed from current observations.

The resulting loop connects memory retrieval, intent generation, world prediction, and experience consolidation. Retrieved knowledge guides the frozen Agent, while world evidence and execution outcomes inform subsequent memory updates. During internal deliberation, new evidence and retrieved experience can refine plans even when the task objective and current observation remain unchanged. Memory thus supports decision refinement without updating the Agent's parameters, complementing World–Action adaptation for improved prediction and execution.

A.5 Data stastic

A.5.1 Hierarchical Decomposition and Statistics of BiCoord

To support the dual-brain paradigm, we segment BiCoord manipulation trajectories into skill-level and primitive-level units, yielding BiCoord-Skill and BiCoord-Primitive, respectively. BiCoord-Skill enriches the Agentic Reasoning Brain's memory with knowledge of the skills required to complete a task, while BiCoord-Primitive provides elementary manipulation units for training the World–Action Brain.

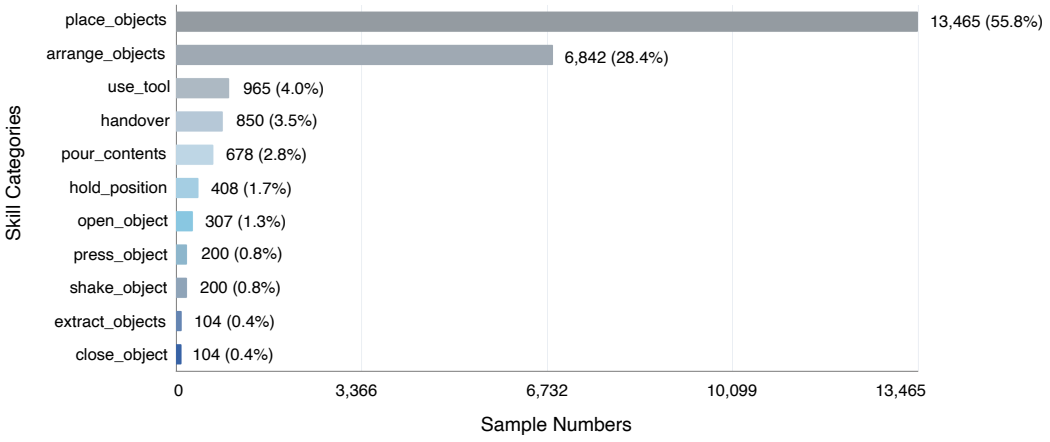


Figure 12: Skill distribution in BiCoord-Skill across 11 manipulation categories. Bar lengths indicate annotation counts, with corresponding percentages shown alongside.

BiCoord Skill. As shown in Fig 12, BiCoord-Skill comprises 11 manipulation skill categories, each representing a task stage with a specific manipulation objective. Object placement and arrangement account for 55.8% and 28.4% of the annotations, respectively, followed by tool use (4.0%), handover (3.5%), and pouring (2.8%). The remaining categories include holding, opening, pressing, shaking,

extraction, and closing. Organized around complete skills, this representation supports the Agentic Reasoning Brain in tracking task progress and selecting subsequent operations.

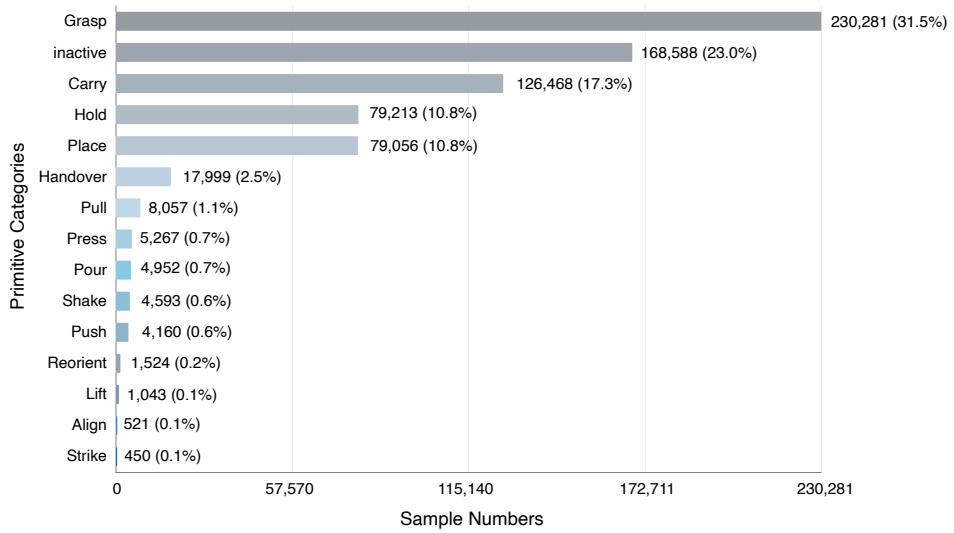


Figure 13: Primitive distribution in BiCoord-Primitive across 15 action and state categories. Bar lengths indicate annotation counts, with corresponding percentages shown alongside.

BiCoord Primitive. As shown in Fig 13, BiCoord-Primitive further decomposes skill execution into 15 categories of action primitives and states. Grasping accounts for 31.5% of the annotations, inactivity for 23.0%, carrying for 17.3%, and holding and placing for 10.8% each. The remaining categories include handover, pulling, pressing, pouring, shaking, pushing, reorientation, lifting, alignment, and striking. This representation supports physical interaction modeling and primitive-conditioned action generation in the World–Action Brain.

A.5.2 Hierarchical Decomposition and Statistics of RoboDojo

Following the same hierarchical decomposition, we segment RoboDojo trajectories into skill-level and primitive-level units, yielding RoboDojo-Skill and RoboDojo-Primitive, respectively, to support high-level task reasoning and low-level action execution.

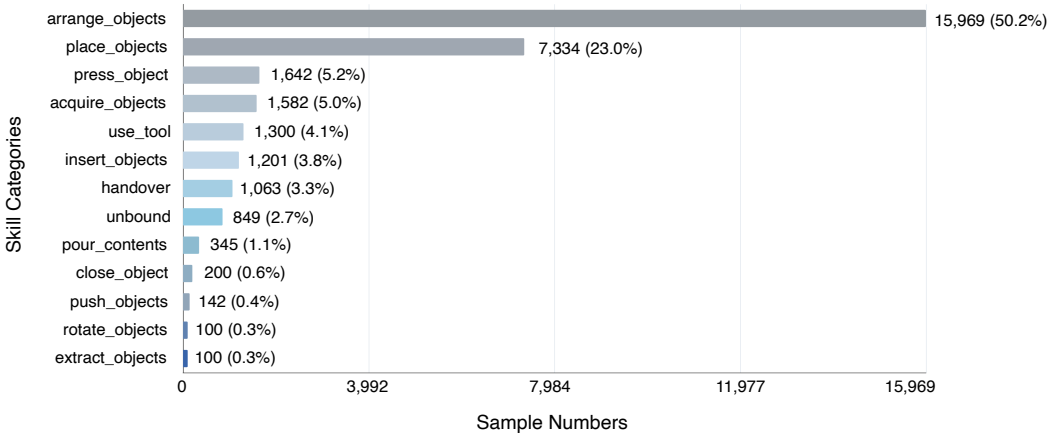


Figure 14: Skill distribution in RoboDojo-Skill across 13 annotation categories. Bar lengths indicate annotation counts, with corresponding percentages shown alongside.

RoboDojo Skill. As shown in Fig 14, RoboDojo-Skill contains 13 annotation categories. Object arrangement and placement dominate the distribution, accounting for 50.2% and 23.0% of the annotations, respectively. Pressing, object acquisition, tool use, insertion, and handover account for 5.2%,

5.0%, 4.1%, 3.8%, and 3.3%, respectively. The remaining categories include `unbound`, pouring, closing, pushing, rotation, and extraction. This skill-level representation organizes trajectories into manipulation stages, supporting the Agentic Reasoning Brain in tracking task progress and selecting subsequent operations.

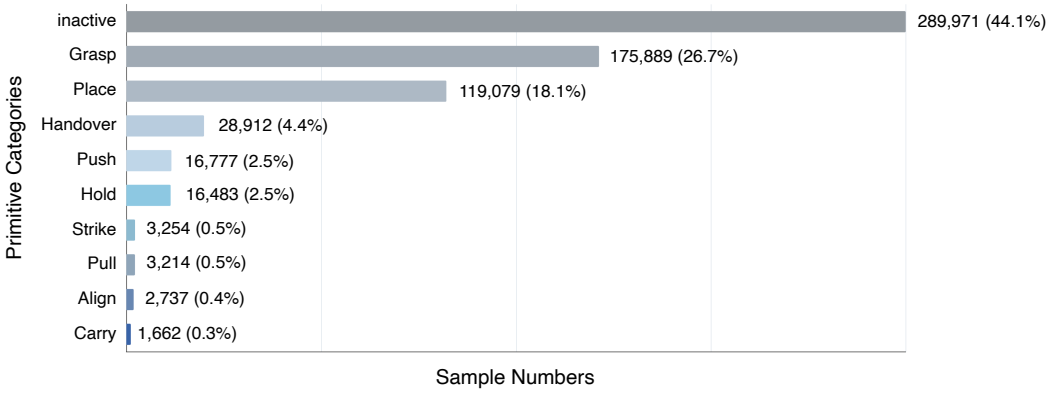


Figure 15: Primitive distribution in RoboDojo-Primitive across 10 action and state categories. Bar lengths indicate annotation counts, with corresponding percentages shown alongside.

Primitive-RoboDojo. As shown in Fig 15, RoboDojo-Primitive comprises 10 categories of action primitives and states. Inactivity accounts for 44.1% of the annotations, followed by grasping (26.7%) and placing (18.1%). Handover accounts for 4.4%, while pushing and holding each account for 2.5%. The remaining categories include striking, pulling, alignment, and carrying, each contributing no more than 0.5%. This fine-grained representation captures within-skill execution, supporting physical interaction modeling and primitive-conditioned action generation in the World–Action Brain.

B Training and Inference Algorithms

This section presents the training and inference algorithms of AWAM. Multi-intent rollout training integrates world prediction with simulator feedback to update the World–Action Brain, Physical Head, and experience memory. Adaptive inference generates a single intent by default and selectively invokes further deliberation based on predicted values and execution feedback. The parameters of the Agentic Reasoning Brain remain frozen throughout.

B.1 World Imagination Training

Algorithm 1 summarizes the training procedure. For each sampled decision context, AWAM performs J deliberation rounds. In each round, the frozen π_A proposes K intents using retrieved experience and feedback from preceding rounds. The World–Action Brain predicts their consequences and generates actions, while V_ψ estimates their returns. Candidates are evaluated from the same simulator snapshot under the specified continuation policy to obtain $G_t^{j,k}$. The resulting records are used to update experience memory and provide feedback for the next round.

Across rounds, AWAM collects value-training records $\mathcal{R}_t$ and constructs within-round preference pairs $\mathcal{P}_t$ for return regression and ranking, respectively. Observed future frames provide world-prediction targets, while demonstrations and eligible successful execution segments provide action targets. The parameters of π_W and V_ψ are shared across rounds and optimized with AdamW, while π_A remains frozen.

B.2 Adaptive Inference

Algorithm 2 summarizes the inference procedure. The Agentic Reasoning Brain retrieves experience, decomposes the task into an ordered subtask sequence, and specifies the primitive invocation

Algorithm 1 Multi-Intent Rollout Training with FPO++

Input: Frozen π_A; world–action model π_W; value estimator V_ψ; memory $\mathcal{M}$; candidate count K; round budget J.
for each training iteration **do**
  Sample task g and context x_t; save simulator snapshot σ_t
  $\pi_W^{\text{old}} \leftarrow \pi_W$; $\mathcal{B}_t \leftarrow \varnothing$; $\mathbf{e}_t^0 \leftarrow \varnothing$
  for $j = 1, \dots, J$ **do**
    Retrieve m_t from $\mathcal{M}$ and construct $\tilde{x}_t^j = (g, o_t, h_t, m_t, \mathbf{e}_t^{j-1})$
    Generate $\{z_t^{j,k}\}_{k=1}^K$ using $\pi_A(\cdot \mid \tilde{x}_t^j)$
    for $k = 1, \dots, K$ **do**
      Predict $(W_t^{j,k}, \mathbf{e}_t^{j,k})$ and decode $A_t^{j,k}$ using π_W^{old}
      $v_t^{j,k} \leftarrow V_\psi(W_t^{j,k}, g)$
      Restore σ_t and execute $A_t^{j,k}$ to obtain rollout $\tau_t^{j,k}$
      $G_t^{j,k} \leftarrow \sum_{h=1}^H \gamma^{h-1} \rho_{t+h}^{j,k}$
    end for
    Collect contexts, intents, predictions, actions, and outcomes in $\mathcal{B}_t^j$
    $\mathcal{B}_t \leftarrow \mathcal{B}_t \cup \mathcal{B}_t^j$
    Update $\mathcal{M}$ and feedback $\mathbf{e}_t^j$ from $\mathcal{B}_t^j$ *// Experience Consolidation*
  end for
  Compute advantages from $\mathcal{B}_t$ following the FPO++ training protocol
  Construct physical supervision targets from collected interaction outcomes
  for each optimization mini-batch **do**
    Compute $\mathcal{L}_{\text{FPO++}}$ using π_W and fixed π_W^{old}
    Compute $\mathcal{L}_{\text{PHY}}$ using physical supervision targets
    $\mathcal{L}_{\text{online}} \leftarrow \mathcal{L}_{\text{FPO++}} + \lambda'_{\text{PHY}} \mathcal{L}_{\text{PHY}}$
    Update trainable parameters using $\mathcal{L}_{\text{online}}$; keep π_A frozen
  end for
end for

and intent for the active subtask. Conditioned on this intent, the World Brain predicts physical consequences, while V_ψ estimates the associated return. The Action Policy generates actions from the future context $\mathbf{f}_t = \mathcal{F}_\omega(W_t)$, which is cached throughout the generation of the current action chunk.

AWAM continues executing the active subtask across action chunks as needed and advances to the next subtask upon completion. Each action chunk uses a single predicted future without multi-candidate search. Execution outcomes are used to update interaction history and experience memory. The Agentic Reasoning Brain revises the remaining plan only when execution is poor or substantial prediction mismatches occur. The future context is refreshed as observations or intents change. Execution ends upon task completion or exhaustion of the execution budget.

C Theoretical Analysis

C.1 Proof of Proposition 1

We fix a decision step t, a deliberation round j, and a finite candidate set $\mathcal{C}_t^j$. All distributions below are conditioned on the current reasoning context and candidate set. These conditions are omitted for brevity. Let π_A^j be the normalized candidate prior and define its support as $\mathcal{S}_t^j = \{z \in \mathcal{C}_t^j : \pi_A^j(z) > 0\}$. Assume that the estimated execution-success likelihood $\ell_W^j(z)$ satisfies $0 < \ell_W^j(z) \leq 1$ for every $z \in \mathcal{S}_t^j$. The normalizing constant and reference posterior are

$$Z_j = \sum_{z \in \mathcal{S}_t^j} \pi_A^j(z) \ell_W^j(z), \qquad q_j^*(z) = \frac{\pi_A^j(z) \ell_W^j(z)}{Z_j}. \tag{11}$$

The assumptions imply $0 < Z_j \leq 1$. We set $q_j^*(z) = 0$ outside $\mathcal{S}_t^j$.

For any candidate distribution q satisfying $q \ll \pi_A^j$, consider the objective

$$\mathcal{F}_j(q) = \mathbb{E}_{z \sim q}[\log \ell_W^j(z)] - D_{\text{KL}}(q \| \pi_A^j). \tag{12}$$

Algorithm 2 Adaptive Inference

Input: Task g; π_A, π_W, V_ψ; memory $\mathcal{M}$; threshold δ; execution budget.
$t \leftarrow 0$; observe (o_0, q_0); $h_0 \leftarrow \varnothing$; $\mathbf{e}_{\text{fb}} \leftarrow \varnothing$; $b \leftarrow 1$
while task incomplete and execution budget remains **do**
  $m_t \leftarrow \texttt{retrieve}(\mathcal{M}, g, o_t)$; $x_t \leftarrow (g, o_t, h_t, m_t)$
  if $b = 1$ **then**
    $(S, c) \sim \pi_A(\cdot \mid x_t, \mathbf{e}_{\text{fb}})$, where $S = (s_1, \ldots, s_n)$ *// Task Decomposition*
    $i \leftarrow 1$; $b \leftarrow 0$
  end if
  $c_t \leftarrow \texttt{ground_invocation}(s_i, c, x_t)$; $z_t \leftarrow \texttt{encode_intent}(s_i, c_t)$
  Predict $(W_t, \mathbf{e}_t)$ using $\pi_W(\cdot \mid o_t, z_t)$; $v_t \leftarrow V_\psi(W_t, g)$ *// Single World Prediction*
  Cache $\mathbf{f}_t \leftarrow \mathcal{F}_\omega(W_t)$; $A_t \leftarrow \texttt{sample_actions}(\mathbf{f}_t, q_t)$
  $(o_{t+1}, q_{t+1}, u_t) \leftarrow \texttt{execute}(A_t)$
  $d_t \leftarrow \texttt{discrepancy}(W_t, u_t)$; $\mathbf{e}_{\text{fb}} \leftarrow \texttt{feedback}(\mathbf{e}_t, v_t, u_t)$
  Update h_{t+1} and $\mathcal{M}$ with $(x_t, z_t, A_t, \mathbf{e}_t, v_t, u_t)$
  if $\texttt{poor_execution}(u_t)$ or $d_t > \delta$ **then**
    $b \leftarrow 1$ *// Adaptive Replanning*
  else if $\texttt{completed}(s_i, o_{t+1})$ **then**
    $i \leftarrow i + 1$
    if $i \leq n$ **then**
      Select invocation c for s_i using π_A with updated observations and feedback *// Next Subtask*
    else
      $b \leftarrow 1$ *// Replan if Task Remains*
    end if
  end if
  $t \leftarrow t + 1$
end while

Substituting the definition of q_j^* gives

$$\begin{aligned} D_{\mathrm{KL}}(q \| q_j^*) &= \sum_{z \in \mathcal{S}_t^j} q(z) \log \frac{q(z) Z_j}{\pi_A^j(z) \ell_W^j(z)} \\ &= D_{\mathrm{KL}}(q \| \pi_A^j) - \mathbb{E}_{z \sim q}[\log \ell_W^j(z)] + \log Z_j. \end{aligned} \tag{13}$$

Therefore,

$$\mathcal{F}_j(q) = \log Z_j - D_{\mathrm{KL}}(q \| q_j^*). \tag{14}$$

By nonnegativity of the KL divergence, $\mathcal{F}_j(q) \leq \log Z_j$, with equality if and only if $q = q_j^*$. Hence, $q_j^* = \pi_{\mathrm{RW}}^j$ is the unique optimizer of Eq. 5. Moreover, the objective gap is exactly $\mathcal{F}_j(q_j^*) - \mathcal{F}_j(q) = D_{\mathrm{KL}}(q \| q_j^*)$, which completes the proof.

C.2 Assumptions and Scope

The proposition applies to a fixed finite candidate set with a normalized prior and positive execution-success likelihoods on the prior's support. It characterizes the unique variational optimizer for the given prior and likelihood estimates. The likelihood should condition on all information relevant to execution success. The success likelihood differs from $V_\psi(W, g)$, which estimates discounted return and cannot generally be interpreted as a success probability. The posterior serves as a theoretical reference. AWAM refines decisions through world evidence and memory updates without explicitly computing this distribution. The proposition therefore does not guarantee the optimality of the implemented updates, convergence across rounds, or monotonic improvement in task success. Practical gains depend on predictive accuracy and are assessed through execution-level experiments.

D Benchmarks and Evaluation

D.1 RoboDojo

RoboDojo is a unified benchmark for generalist robotic manipulation, comprising 42 simulation tasks and 18 real-world tasks (Chen et al., 2026). Its simulation suite uses the ARX X5 bimanual

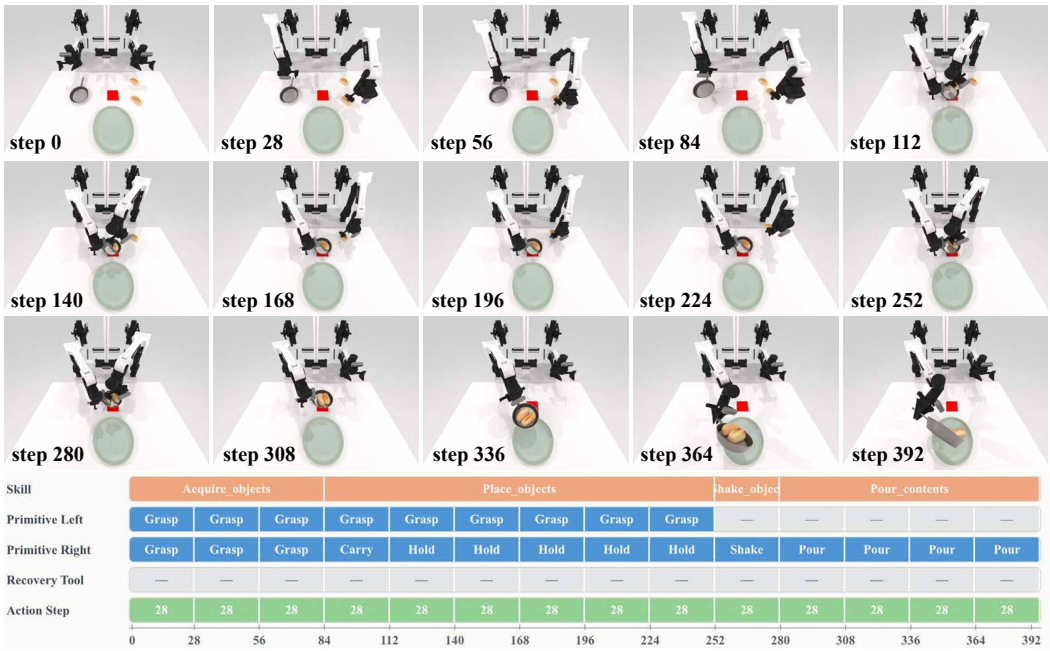


Figure 16: AWAM executing a bimanual cooking task. The robot places food items into a pan, shakes the pan, and pours its contents onto a plate. The timeline aligns task-level skills with per-arm primitives, showing coordinated execution without recovery interventions.

platform in Isaac Sim and supports parallel evaluation across diverse tasks. The accompanying simulation dataset contains 3,500 demonstration trajectories covering 34 training tasks and an auxiliary augmentation set, while the eight Open tasks are reserved for evaluating transfer to unseen task specifications.

Our evaluation covers generalization, precision, long-horizon execution, memory, and open-vocabulary instruction following. These categories assess robustness to environmental changes, fine-grained control, multi-stage completion, retention of historical observations, and skill recombination under novel instructions. Table 1 reports score / success rate for each category, measuring partial task progress and complete task success, respectively. The overall results are computed by averaging the corresponding metrics across the five categories.

D.2 BiCoord

BiCoord is built on RoboTwin 2.0 (Chen et al., 2025) and comprises 18 bimanual manipulation tasks requiring sustained inter-arm coordination and dynamic role exchange across multiple subgoals (Peng et al., 2026). Each task provides 100 successful demonstration trajectories with stage-level annotations of temporal boundaries, subgoals, and arm behaviors. These annotations support evaluation of intermediate progress alongside complete task execution.

We use the multi-task learning setting and report results in Table 2. The benchmark protocol evaluates each task over 100 rollouts using Success Rate (SR), Stage-Wise Success Rate (SSR), and Trajectory Length (TL). SR measures full-task completion, while SSR measures partial progress through a weighted sum of completed stages. TL counts execution steps in successful trajectories. When computing average TL, tasks on which any compared method has zero success are excluded to ensure a common comparison set.

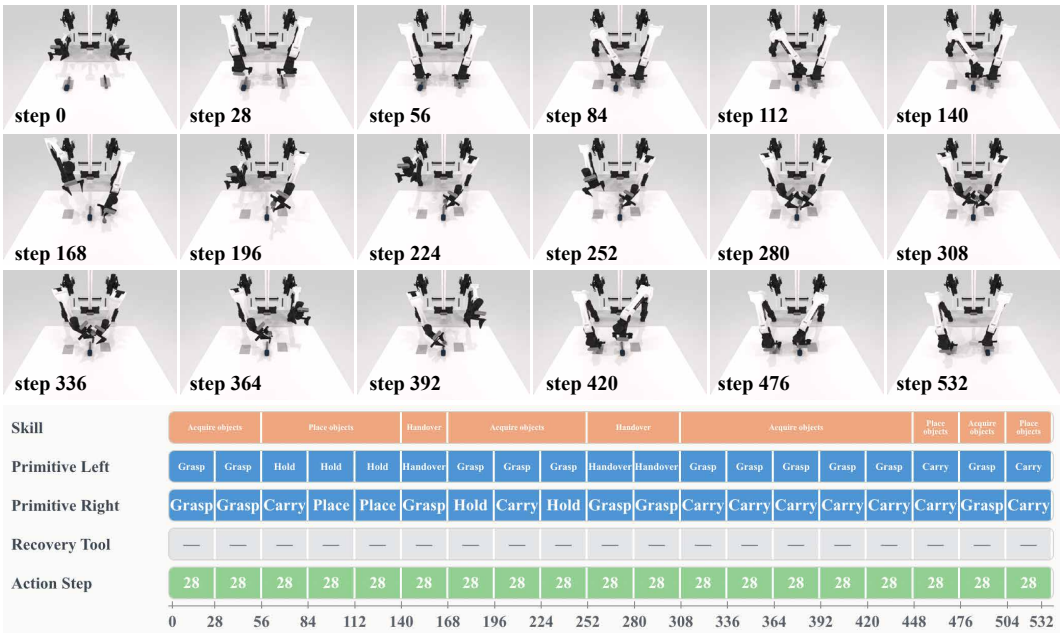


Figure 17: AWAM executing a bimanual microphone exchange task through grasping, inter-arm handovers, and placement. The timeline aligns task-level skills with per-arm primitives, illustrating coordinated execution without recovery interventions.

E Qualitative Results

E.1 Dual-Brain Collaboration in Long-Horizon Tasks

The four examples illustrate how AWAM connects task-level reasoning with coordinated action execution. The Agentic Reasoning Brain organizes each task into skills and per-arm primitive instructions, while the World–Action Brain generates the corresponding actions. The aligned timelines show how these instructions structure execution across successive subgoals and different forms of bimanual coordination.

The block-sweeping and bridge-building tasks illustrate complementary and synchronized arm roles, respectively. During block sweeping, one arm holds the dustpan while the other sweeps the block into it, requiring coordination between holding and manipulation. During bridge building, the robot first positions two support blocks and then uses both arms to place a beam across them. This sequence illustrates the dependency between preparing the supports and assembling the final structure. The microphone exchange task requires repeated changes in arm roles through grasping, carrying, handovers, and placement. AWAM sequences these operations to transfer the microphones between arms and move them to their target locations. The cooking task involves a different composition of skills: placing food into a pan, shaking the pan, and pouring its contents onto a plate. Execution must preserve the order of these stages while coordinating food manipulation and pan handling.

Together, these trajectories illustrate how a shared hierarchy of skills and primitives supports diverse task structures, from simultaneous manipulation to sequential handovers and tool use. No recovery tools are invoked in the displayed runs, providing qualitative examples of sustained dual-brain collaboration across multiple task stages.

We visualize long-horizon execution through synchronized sequences of observations, structured primitives, physical evidence, and execution outcomes. The Agentic Reasoning Brain selects the next manipulation primitive, while the World–Action Brain generates action chunks conditioned on incoming observations. Primitive boundaries mark when high-level reasoning is invoked. These examples illustrate how task-level decisions are translated into physical actions and how execution feedback informs subsequent stages.

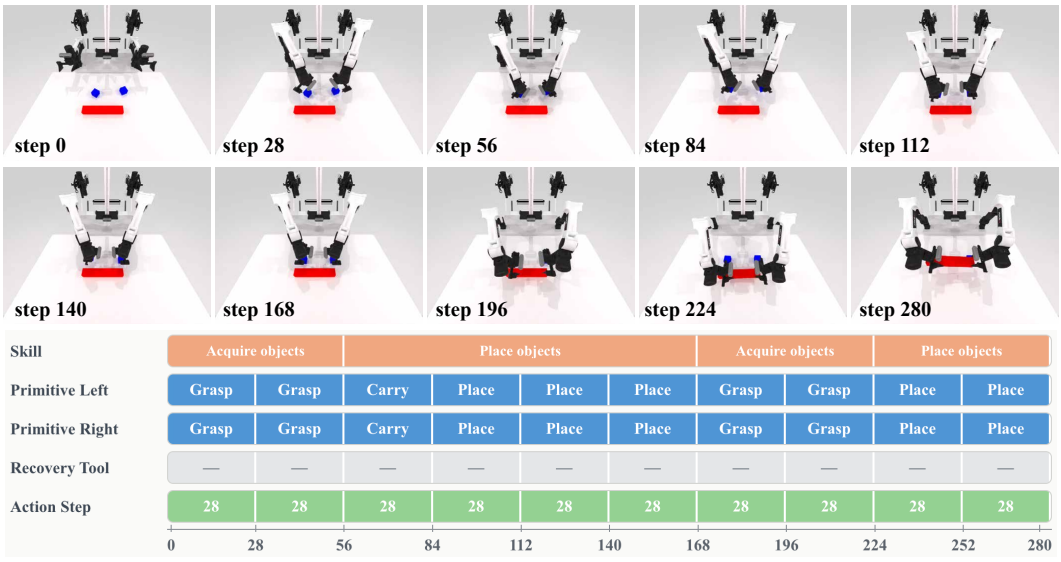


Figure 18: AWAM executing a bimanual bridge-building task. The robot positions two support blocks and jointly places a beam across them. The timeline aligns task-level skills with per-arm primitives, illustrating synchronized execution without recovery interventions.

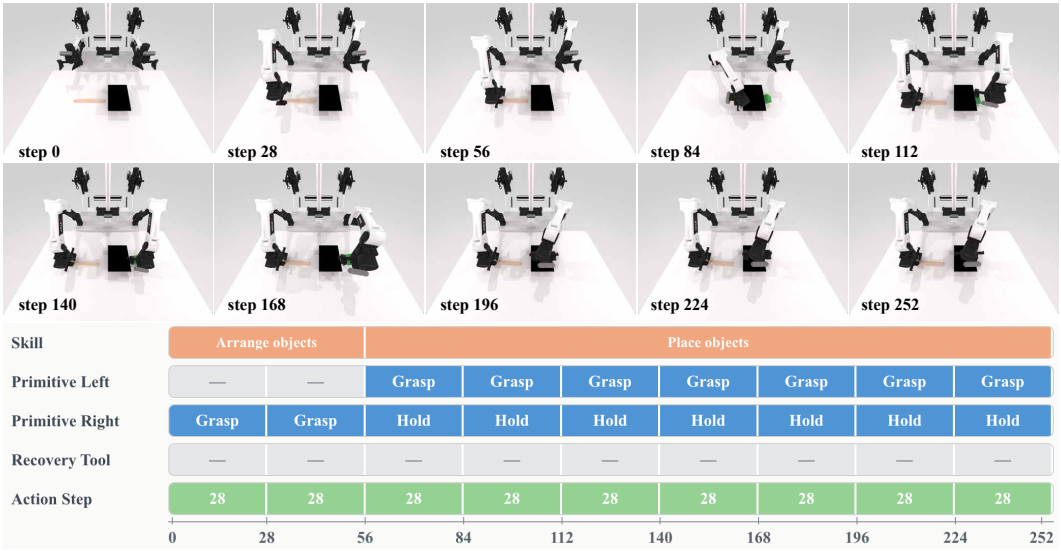


Figure 19: AWAM executing a bimanual block-sweeping task. One arm holds the dustpan while the other sweeps the block into it. The timeline aligns task-level skills with per-arm primitives, illustrating complementary arm roles without recovery interventions.

E.2 Recovery from Execution Failures

We visualize AWAM's recovery behavior during a bimanual pen-collection task. Following an execution failure, AWAM invokes a recovery tool and resumes collecting and placing pens into the container. The highlighted frame and timeline indicate when recovery occurs within the ongoing skill sequence.

This example illustrates how AWAM incorporates recovery into execution while preserving the overall task goal. The transition from regular primitives to recovery and back to task execution provides qualitative evidence of its ability to recover from a local failure and resume progress.

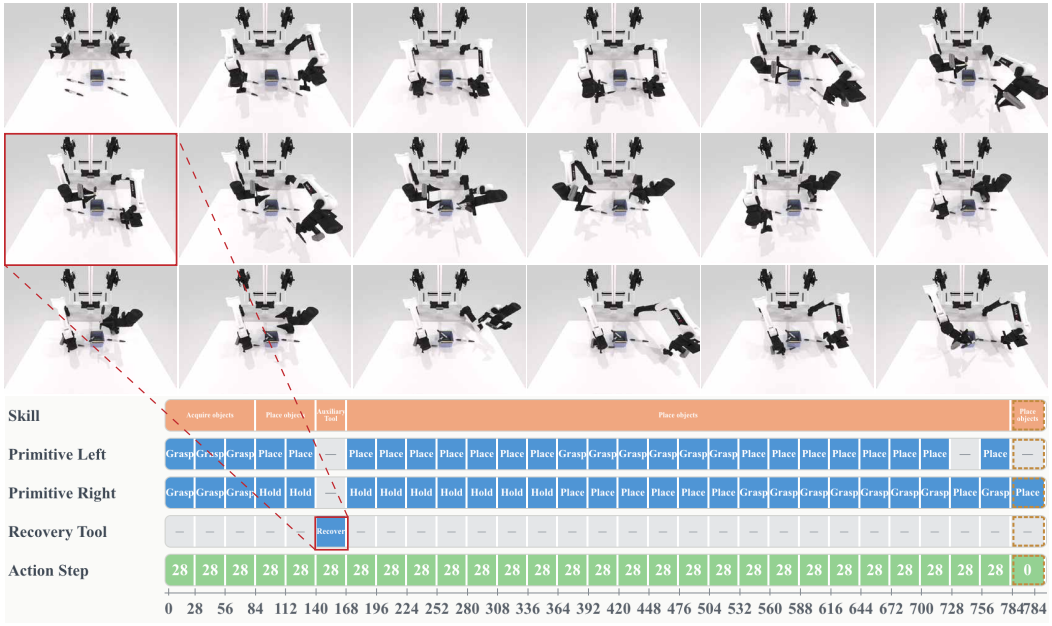


Figure 20: Failure recovery during a bimanual pen-collection task. AWAM invokes a recovery tool following an execution failure and resumes collecting and placing pens into the container. Red annotations link the recovery event to the corresponding execution frame and timeline.

Table 6: Real-world performance on two manipulation tasks. Task entries report successes out of 30 trials and success rates (%). Avg. denotes the mean success rate across tasks.

Method	**Real-World Tasks**		**Avg. (%)**
	Block Stacking (%)	Number Sorting (%)	
$\pi_{0.5}$	73.3	60	66.65
AWAM (Ours)	83.4	76.6	80

E.3 Real-world Experiments

We evaluate AWAM on two tabletop manipulation tasks using the Piper X robotic platform from AgileX Robotics: *Block Stacking* and *Number Sorting*. Block Stacking requires placing a red block on a yellow block, while Number Sorting requires arranging numbered blocks in the instructed order. We compare AWAM with $\pi_{0.5}$ Black et al. (2025) under the same hardware, observation inputs, task instructions, and execution budgets.

Each method is evaluated over 30 trials per task, yielding 120 trials in total. Initial object positions are randomized, with the same set of initial layouts used across methods. Model parameters remain fixed during evaluation. A trial succeeds when all task requirements are satisfied within the execution budget without human assistance. Table 6 reports success rates, while execution visualizations illustrate AWAM's object identification, spatial reasoning, and sequential manipulation.

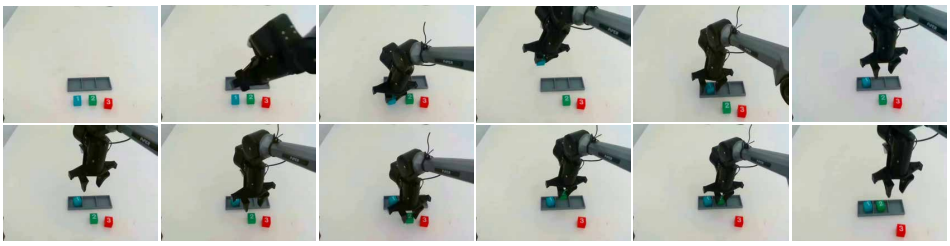


Figure 21: Real-world execution of the number-sorting task with AWAM. The robot sequentially places numbered blocks into a tray in the instructed order. Orange and blue bars indicate placement skills and their corresponding grasping and placing primitives, respectively.