WA*R*M: WORLD-ACTION-RETROSPECTION MODEL FOR *C*ONSEQUENCE-*A*LIGNED *R*EUSE OF *E*XPERIENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Local-context world–action models (WAMs) couple visual prediction with control but leave task-critical earlier interactions outside their decision context. Memory-augmented embodied policies extend access to history, yet retaining past information does not establish whether a recalled action remains appropriate. The central challenge is to use ongoing interaction as evidence for deciding which past behavior to reuse and how it should shape action generation. We introduce WA*R*M ***(World–Action–Retrospection Model)***, organized around *event-grounded retrospection*: current context and *situated recollection* ground inference of a candidate-independent transition requirement, while a cross-episode *sensorimotor repertoire* supplies state–action–effect events. Adapted repertoire actions are assessed by comparing their predicted effects with this learned forecast; *consequence-gated source transport* then jointly regulates proposal-informed stochastic initialization and candidate-derived conditioning, turning recalled behavior into a revisable prior for flow-based action generation. WA*R*M achieves 98.7% success on LIBERO and 83.9% on RMBench, including 87.3% on tasks requiring multiple past observations. Component comparisons suggest complementary benefits from recollection and repertoire; complete-event distraction tests show less degradation with explicit consequence comparison. These findings support using current evidence to govern both the selection of past behavior and its influence on action generation.

1 INTRODUCTION

Successful long-horizon manipulation requires connecting what has already happened to what should happen next. World–action models (WAMs) couple visual prediction with action generation, bringing predictive representations of physical interaction into robot control (Kim et al., 2026; Li et al., 2026; Bi et al., 2025). Yet understanding how a scene can change does not determine which change the current task calls for, or which past behavior is worth reusing.

Existing approaches address different parts of this connection. *Local-context WAMs* act from current observations or short histories (Figure 1a), which can omit earlier interactions needed to disambiguate the next action (Kaelbling et al., 1998; Dai et al., 2026; Lei et al., 2026). *Memory-augmented WAMs* retain historical context or derive memory-grounded plans (Figure 1b) (Yang et al., 2026b; Wang et al., 2026; Zhao et al., 2026). Recent VLA policies also combine online visual history with retrieved experience prompts (Zhou et al., 2026a). Access to either alone does not establish a recalled action's present applicability. *Trajectory retrieval and informed flow sources* supply action exemplars and structured generation priors (Park et al., 2026; Kang et al., 2026; Jia et al., 2026; Machado et al., 2026); a historically successful action, however, may repeat a completed interaction or produce an unwanted change. These approaches raise a common question: ***how can a WAM infer the current task requirement, identify applicable past actions, and regulate their influence on generation?***

Our perspective echoes the *Analects*' idea of *gaining new insight by revisiting prior knowledge* (Confucius, 2002, 2.11). *Recollection* retains observed changes and executed-action summaries from the current episode; a *repertoire* preserves past behaviors and their observed effects across episodes. The former informs current task requirements; the latter offers reusable action structure rather than merely additional context. Familiar behavior alone cannot reveal what has already happened here, so its applicability must be reassessed against current evidence. Our insight is to use *recollection as evidence for assessing the repertoire, and recalled behavior as a revisable action prior*.

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.

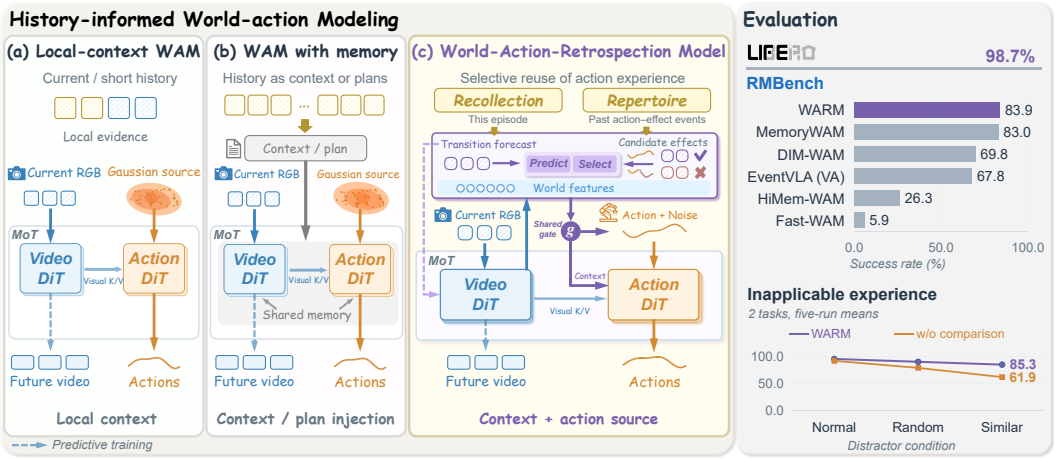


Figure 1: **From interaction evidence to selective action reuse.** (a,b) Illustrative WAM interfaces use local observations or retained history. (c) Recollection grounds the task requirement; the repertoire supplies action–effect proposals. Predicted consequence compatibility informs selection and a shared gate over source and conditioning. K/V denotes visual keys and values; dashed arrows indicate training-only prediction. **Right:** Benchmark results with published references (Tables 1–2) and distraction on Put Back Block and Press Button, averaged equally within each of five evaluation runs.

We introduce WA*R*M, the *World-Action-Retrospection Model*, a video–action architecture for *event-grounded retrospection* (Figure 1c). Current world features and *situated recollection* ground the task requirement; the *sensorimotor repertoire* supplies cross-episode state–action–effect events as reusable proposals. We adapt and assess these proposals by predicted effect, then use *consequence-gated source transport* to regulate their contribution to Action DiT initialization and conditioning.

Consequence-aligned action recall compares the required change with each recalled action's predicted effect. We infer a candidate-independent requirement from current context and recollection, and predict proposal effects from adapted actions, world features, and stored effect evidence. Compatibility complements history-aware ranking and guides selection and gating.

Consequence-gated source transport carries this assessment into generation. A shared gate jointly regulates proposal-informed stochastic initialization and candidate-derived conditioning, letting Action DiT refine useful action structure as a revisable prior. Closing the gate restores Gaussian initialization while preserving recollection-grounded task conditioning. This shared control admits, attenuates, or withdraws recalled behavior without iterative future-video generation.

Our contributions are threefold:

- We introduce event-grounded retrospection for WAMs, linking history-dependent task understanding to selective reuse of cross-episode action experience.
- We develop consequence-aligned action recall and consequence-gated source transport: a candidate-independent requirement guides proposal assessment, while a shared gate controls initialization and conditioning for selective generative refinement.
- WA*R*M achieves 98.7% success on LIBERO (Liu et al., 2023a) and 83.9% on RM-Bench (Chen et al., 2026), including 87.3% on $M(n)$ tasks requiring multiple past observations. Component results suggest complementary benefits of recollection and repertoire; complete-event distraction tests support explicit comparison with partial event replacement.

2 Related Work

World–action modeling. Predictive models connect visual dynamics to control through several interfaces. Video-planning methods synthesize visual trajectories and translate them into actions (Du et al., 2023), while predictive policies learn control from future-observation representations and inverse dynamics (Tian et al., 2025; Hu et al., 2025). Joint video–action generation couples both

modalities within a generative model (Zhu et al., 2025; Bi et al., 2025; Kim et al., 2026; Ye et al., 2026), with native pretraining extending this coupling (Zhang et al., 2026). Efficient interfaces retain predictive training while omitting future-video generation (Yuan et al., 2026), or expose compact transition representations to action decoding (Ma et al., 2026). These interfaces improve how prediction supports control. WA*R*M addresses the complementary reuse problem by explicitly comparing what a recalled action is predicted to change with the transition required now.

History and experience for long-horizon control. History can enter control as compressed state, selected evidence, or planning context. State-space diffusion policies compress the observation stream into recurrent context (Guan et al., 2026); VLA policies combine perceptual and semantic histories at multiple scales (Shi et al., 2025; Torne et al., 2026). In WAMs, persistent representations, diverse-event selection, and hierarchical summaries extend access to earlier observations (Yang et al., 2026b; Wang et al., 2026; Sun et al., 2026). Action-effect pretraining and prospective working states further align history with control (Zhou et al., 2026b; Guo et al., 2026). MemER and Mem-0 use keyframes or completed interactions for planning (Sridhar et al., 2025; Chen et al., 2026). These methods recover task context. WA*R*M uses episode-specific recollection to infer a candidate-independent requirement and assess the present applicability of repertoire actions.

Action generation and experience-informed priors. Action policies span autoregressive token prediction (Kim et al., 2025), latent-variable action chunking (Zhao et al., 2023), and continuous diffusion or flow generation (Chi et al., 2023; Black et al., 2025). ReCAP conditions on retrieved trajectories and learns residual action corrections, with predictive supervision in its WAM variant (Park et al., 2026). Informed-source flow policies initialize generation with recent actions, proprioceptive histories, or learned action priors (Kang et al., 2026; Jia et al., 2026; Machado et al., 2026). These approaches provide useful action priors. WA*R*M couples consequence assessment to a shared source-and-conditioning gate, so recalled behavior can be admitted, attenuated, or withdrawn.

3 WA*R*M: Consequence-Aligned Retrospection

Overview. We organize WA*R*M around three decisions: infer what the current task requires, assess which past action can help, and regulate its influence on generation. At replan t, the policy observes RGB images o_t, robot state q_t, and instruction ℓ, then generates $A_t = (a_t, \dots, a_{t+H-1}) \in \mathbb{R}^{H \times d_a}$, where H is the prediction horizon and d_a the action dimension, and executes its first $h_{\text{exec}} \leq H$ commands. Our Fast-WAM-derived video–action architecture (Yuan et al., 2026) uses one current-frame video pass to supply retrospective features and visual keys/values (K/V) reused by the transformer action decoder, Action DiT (Peebles & Xie, 2023), during generation. Situated recollection $\mathcal{M}_t$ provides episode-specific evidence, while the sensorimotor repertoire $\mathcal{B}$ provides cross-episode action exemplars. Their separation lets current interaction evidence judge past behavior before generation, with the notation and module outputs indexed in Appendix A.6.

3.1 Event-Grounded Recollection and Repertoire

Retain evidence from this episode. Situated recollection $\mathcal{M}_t$ contains the initial observation, selected interaction records, and summaries of action prefixes already executed in the current episode. Visual change, gripper transitions, and commanded-direction changes trigger retention; the initial and latest records are protected, while redundant interior records are merged when capacity is reached. Training reconstructs the same past-only trace before each action chunk, grounding requirement inference in observed interaction; Appendix A.5 gives the retention, compression, and readout rules.

Preserve reusable actions with their effects. We build the sensorimotor repertoire $\mathcal{B}$ from complete H-step windows in training demonstrations. With frozen visual encoder Φ and window start s_i, each event stores pre-state features S_i^-, executed actions A_i, and observed feature change ΔS_i:

$$e_i = (S_i^-, A_i, \Delta S_i; k_i, \eta_i), \qquad S_i^- = \Phi(o_{s_i}), \quad \Delta S_i = \Phi(o_{s_i+H}) - S_i^-. \tag{1}$$

Here k_i is the retrieval key and η_i records an action-space start reference, timing, and provenance. A fixed pooling map yields the observed-effect descriptor $E_i = \mathcal{P}_{\text{eff}}(S_i^-, \Delta S_i) \in \mathbb{R}^{d_s}$, where d_s is the semantic feature width (Appendix A.1). At evaluation, $\mathcal{B}$ remains fixed and offline retrieval excludes

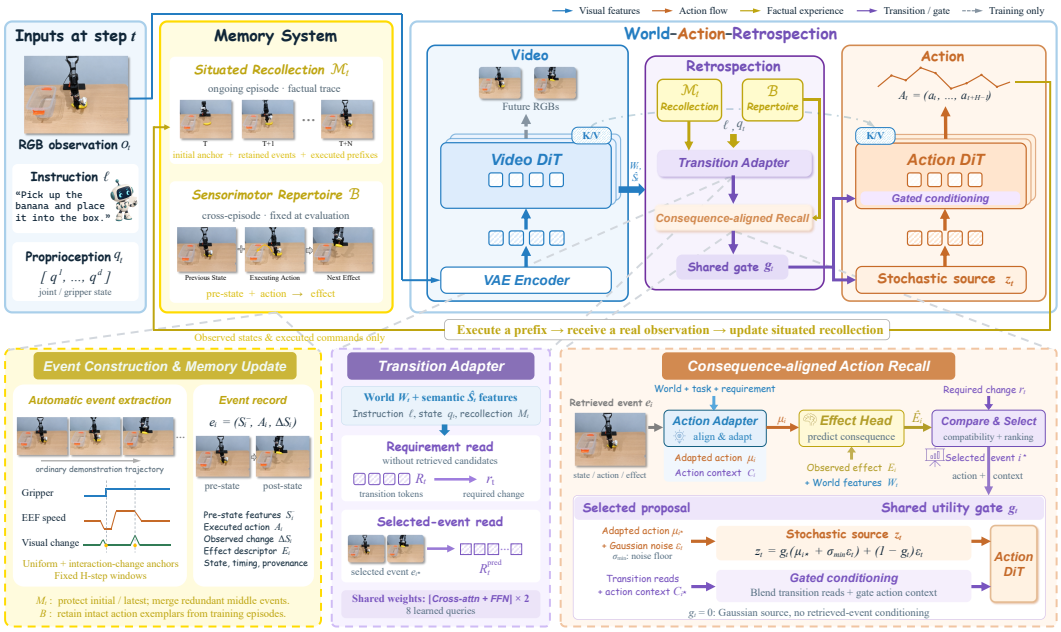


Figure 2: **From interaction evidence to generative action reuse.** Recollection $\mathcal{M}_t$ grounds a transition requirement r_t; repertoire actions are adapted into proposals μ_i and assessed through predicted effects $\widehat{E}_i$. A shared gate g_t regulates the selected proposal in both initialization and conditioning, while executed actions and observations update recollection. The lower panels detail event construction, transition inference, and consequence-aligned recall; dashed arrows denote future-video prediction used only during training to retain predictive supervision.

the query episode, preserving the separation between the episode evidence used to infer requirements and the earlier interactions retrieved from the repertoire for action reuse.

3.2 Recollection-Grounded Transition Inference

We first convert current evidence into a common reference for action reuse. A learned semantic bridge fuses current-frame token features from Video DiT layers 9 and 19, each of width 3,072, into world tokens W_t and compact semantic tokens $\widehat{S}_t$ aligned to $\Phi(o_t)$. Our *Transition Adapter* G_{tr} reads these features with q_t, ℓ, and $\mathcal{M}_t$; after pooling, P_{req} maps its requirement tokens R_t to the effect-space vector r_t used to assess all retrieved action proposals:

$$R_t = G_{\text{tr}}(W_t, \widehat{S}_t, q_t, \ell, \mathcal{M}_t; \varnothing), \qquad r_t = P_{\text{req}}(\text{mean}(R_t)) \,. \tag{2}$$

The masked retrieved-event input $\varnothing$ makes this read independent of the current candidate set. During training, the requirement is supervised by the effect of the demonstrated interaction:

$$E_t^\star = \mathcal{P}_{\text{eff}}(\Phi(o_t), \Phi(o_{t+H}) - \Phi(o_t)) \,. \tag{3}$$

The resulting *required transition* is a task-conditioned forecast of what should change. At deployment, the same r_t serves as the candidate-independent criterion for proposal assessment.

3.3 Consequence-Aligned Action Recall

Retrieve and adapt action proposals. We retrieve a valid candidate set $\mathcal{I}_t$ by cosine top-K search over $\mathcal{B}$ with frozen visual features and closed-set task identity (Appendix A.2). For each event, current and stored action-coordinate starts c_t, c_i^0 determine valid alignment $\mathcal{W}$; event timing τ_i then enters the learned *Action Adapter* f_A, which applies a bounded task-conditioned residual correction to produce the adapted proposal μ_i for subsequent consequence assessment:

$$A_i^w = \mathcal{W}(A_i, c_t, c_i^0), \qquad \mu_i = A_i^w + \rho s \odot \tanh f_A(A_i^w, \tau_i, W_t, R_t, q_t, \ell). \tag{4}$$

The fixed scales $s \in \mathbb{R}^{d_a}_{>0}$ and bound $\rho > 0$ enforce $|\mu_{i,h,d} - A^w_{i,h,d}| \le \rho s_d$. The bounded correction preserves recorded action structure before consequence assessment and refinement.

Estimate and compare effects. Our *effect head* f_E predicts the current-context effect of each adapted proposal from its action statistics, stored effect evidence, and pooled world features:

$$\widehat{E}_i = f_E(\mathrm{stats}(\mu_i), E_i, \mathrm{pool}(W_t)) \,. \tag{5}$$

Here stats summarizes mean, terminal command, and temporal variation, and pool aggregates world tokens. Historical effect E_i anchors the estimate, while μ_i and W_t adapt it to the current interaction. We score proposal–requirement agreement in direction and magnitude by

$$\kappa_i = \cos(\widehat{E}_i, r_t) - \lambda_m \left| \log(\|\widehat{E}_i\|_2 + \epsilon) - \log(\|r_t\|_2 + \epsilon) \right| . \tag{6}$$

Here $\lambda_m \geq 0$ weights magnitude mismatch and $\epsilon > 0$ stabilizes the logarithms and cosine denominator (Appendix A.2). The score combines effect direction and magnitude to assess whether a proposal is predicted to produce the change required by the current task and episode.

Select a proposal in context. Contextual relevance is supplied by a history-aware reranker with score ℓ_i^{rank}, while recorded event order contributes continuity prior $b_{t,i}$. Combining contextual relevance, consequence compatibility, and continuity gives the selection rule

$$\ell_{t,i}^{\mathrm{sel}} = \ell_i^{\mathrm{rank}} + \lambda_c \kappa_i + b_{t,i}, \qquad i^\star = \arg\max_{i \in \mathcal{I}_t} \ell_{t,i}^{\mathrm{sel}}. \tag{7}$$

The coefficient $\lambda_c \geq 0$ weights explicit compatibility. The selected proposal $\mu_{i^\star}$ then passes to the influence gate, while empty retrieval uses the candidate-null zero-gate path.

3.4 Consequence-Gated Source Transport

Control the proposal's contribution. Our gate f_g maps selection confidence, consequence compatibility, adaptation size, support, and executed-action repetition to raw utility $\alpha_t \in (0, 1)$; eligibility and repetition rules then produce the effective gate $g_t \in [0, 1]$ used by both generation pathways (Eqs. (17)–(18)). Actions, proposals, and noise below use normalized control coordinates; generated actions are mapped back to native commands for execution. Given selected proposal $\mu_{i^\star}$ and action-shaped noise $\varepsilon_t \sim \mathcal{N}(0, I)$, the effective gate defines Action DiT's proposal-informed stochastic source, which initializes the complete action chunk refined by the decoder:

$$z_t = g_t(\mu_{i^\star} + \sigma_{\min}\varepsilon_t) + (1 - g_t)\varepsilon_t, \qquad 0 < \sigma_{\min} \leq 1. \tag{8}$$

For fixed proposal and gate, z_t has mean $g_t\mu_{i^\star}$ and noise scale $1 - g_t + g_t\sigma_{\min}$. Thus g_t jointly controls the recalled-action prior and the randomness available for generative refinement.

Apply the same gate to conditioning. A second read of G_{tr} incorporates the selected event into candidate-aware transition tokens R_t^{pred}; adaptation and effect features form action-context tokens $C_{i^\star}$. The same effective gate modulates both candidate-dependent conditioning paths:

$$R_t^{\mathrm{act}} = R_t + \mathrm{sg}(g_t)(R_t^{\mathrm{pred}} - R_t), \qquad C_t^{\mathrm{act}} = \mathrm{sg}(g_t)C_{i^\star}. \tag{9}$$

The stop-gradient sg preserves forward values during training. Together they augment native context and visual K/V to form decoder condition $\mathcal{C}_t$, fixed throughout the current replan.

Generate with a revisable action prior. Action DiT integrates $v_\theta(x_\sigma, \sigma; \mathcal{C}_t)$ from $x_1 = z_t$ to $A_t = x_0$ along decreasing flow time σ. Because recalled experience enters through both source and conditioning, the shared gate gives both pathways a common withdrawal point while preserving recollection-grounded task evidence. This makes admission strength an explicit part of the generative interface and ties proposal selection to the amount of candidate influence.

Proposition 1 (Joint withdrawal of recalled experience) *For one replan, fix factual inputs, model parameters, conditioning masks, solver schedule, and sampling randomness. Fixing $g_t = 0$ makes every solver iterate and the generated action invariant to replacement of the retrieved candidates.*

Thus rejection removes selected experience from current generation while retaining episode-specific evidence. Appendix A.3.1 extends the exact zero-gate result: under fixed-branch regularity conditions, $\|A_t(g) - A_t(0)\| \leq gK_t$ bounds partial influence as the gate approaches the candidate-null path.

3.5 Learning and Causal Execution

Learn refinement from the chosen source. Each demonstrated action $A_t^\star$ is paired with its query-conditioned source z_t. Conditional flow matching (Lipman et al., 2023; Liu et al., 2023b) defines the interpolation and target velocity for these experience-conditioned endpoints:

$$x_\sigma = (1-\sigma)A_t^\star + \sigma\,\mathrm{sg}(z_t), \qquad v^\star = \mathrm{sg}(z_t) - A_t^\star. \tag{10}$$

Using valid-action distance d_A and scheduler weight $w(\sigma)$, the action-flow objective is

$$\mathcal{L}^a_{\mathrm{FM}} = \mathbb{E}_{t,\varepsilon,\sigma}\left[w(\sigma)d_A(v_\theta(x_\sigma,\sigma;\mathcal{C}_t),v^\star)\right]. \tag{11}$$

Inference integrates the same velocity field from the same source construction. Proposition 2 in Appendix A.3.3 gives the population-optimal regression field for these experience-dependent source–action pairs, linking selective admission to generative refinement.

Supervise the decisions that shape generation. Auxiliary objectives align the semantic bridge and supervise requirement and effect prediction, ranking, adaptation, and raw utility. Together with video-flow loss $\mathcal{L}^v_{\mathrm{FM}}$, these objectives form the shared training loss used for WA*R*M:

$$\mathcal{L} = \mathcal{L}^a_{\mathrm{FM}} + \lambda_v\mathcal{L}^v_{\mathrm{FM}} + \sum_{j\in\mathcal{J}}\lambda_j\mathcal{L}_j, \qquad \mathcal{J} = \{\mathrm{bridge},\mathrm{req},\mathrm{effect},\mathrm{rank},\mathrm{gate},\mathrm{adapt}\}. \tag{12}$$

The nonnegative coefficients λ_v, λ_j balance these terms. Detaching z_t fixes flow endpoints within an update while conditioning retains permitted gradients; demonstrated futures enter through an isolated video-only pass. Appendix A.4 specifies auxiliary targets, corruptions, and gradient routes.

Reassess after real interaction. After executing the first h_{exec} commands, their summaries and observed feedback update $\mathcal{M}_t$ for the next replan. With parameters and $\mathcal{B}$ fixed at evaluation, each new observation refreshes the required transition, proposal assessment, and gate. Realized interaction thus becomes fresh evidence for weighting recalled experience at the next replan.

4 Experiments

We ask four questions linking task performance to retrospective action reuse. **Q1:** How well does WA*R*M perform in simulation and physical manipulation? **Q2:** What do recollection, repertoire, and their use in action generation each contribute? **Q3:** Can consequence-based assessment preserve the benefit of experience when retrieval is misleading? **Q4:** Do predicted effects distinguish candidate outcomes and help select actions appropriate to the current task?

4.1 Experimental Setup

Benchmarks and metrics. We evaluate all 40 tasks in LIBERO-Spatial, Object, Goal, and Long (Liu et al., 2023a), and nine history-dependent tasks in RMBench (Chen et al., 2026). RMBench has five $M(1)$ tasks requiring one task-relevant past observation and four $M(n)$ tasks requiring multiple past observations. It uses 50 expert demonstrations per task; evaluation comprises 100 rollouts per task on RMBench and 50 on LIBERO. We report task-averaged success rate (SR, %), including failures in each task denominator; the physical protocol specifies 20 trials per method/task.

Implementation and comparisons. At each replan, our video–action mixture of transformers assesses proposals using current-frame world features and conditions an $H = 32$ action chunk on visual keys and values. LIBERO uses two RGB views, seven-dimensional actions, and a ten-step execution prefix; RMBench uses head and dual-wrist views, 14-dimensional joint commands, and a four-step prefix. Observed feedback and executed actions update recollection; the training-episode repertoire stays fixed. Our PiPER arm uses head and wrist RGB views. Benchmark protocols appear in Appendix B; physical deployment and tasks in Appendices C.3.1–C.3.2.

4.2 Simulation Benchmark Results (Q1)

Consistently strong manipulation on LIBERO. WA*R*M reaches 98.7% average SR across LIBERO's four suites, the highest among the listed published references, with each suite at or

Table 1: **LIBERO success rates (%).** Across comparative tables, bold/underlining indicate the best/second-best values for each metric and setting, including ties. References retain published protocols; WLA-0 uses standard inference. Details appear in Appendix B.

Method	Spatial ↑	Object ↑	Goal ↑	Long ↑	Avg. ↑
π_0 (Black et al., 2025)	96.8	98.8	95.8	85.2	94.1
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	<u>98.8</u>	98.2	98.0	92.4	96.9
Motus (Bi et al., 2025)	96.8	<u>99.8</u>	96.6	97.6	97.7
HiMem-WAM (Sun et al., 2026)	98.2	<u>99.8</u>	<u>98.4</u>	94.5	97.7
Fast-WAM (Yuan et al., 2026)	98.2	**100.0**	97.0	95.2	97.6
LingBot-VA (Li et al., 2026)	98.5	99.6	97.2	**98.5**	98.5
WLA-0 (Yang et al., 2026c)	**99.0**	**100.0**	97.8	97.6	<u>98.6</u>
WA*R*M	98.4	**100.0**	**98.6**	<u>97.8</u>	**98.7**

Table 2: **RMBench success rates (%).** TMC: task memory complexity; VA: visual anchors. Dashes indicate unavailable results. Details are specified in Appendix B.

Task	TMC	Success rate (%) ↑							
		$\pi_{0.5}$	Fast-WAM	HiMem-WAM	DIM-WAM	EventVLA (VA)	Memory-WAM	WLA-0	WA*R*M (Ours)
Observe and Pick Up	$M(1)$	9	0	**28**	13	21	<u>27</u>	–	25
Rearrange Blocks	$M(1)$	13	0	33	<u>99</u>	96	**100**	–	87
Put Back Block	$M(1)$	11	0	32	<u>98</u>	95	**100**	–	**100**
Swap Blocks	$M(1)$	24	0	38	96	96	**100**	–	<u>98</u>
Swap T	$M(1)$	15	7	27	**97**	87	94	–	<u>96</u>
Average	$M(1)$	14.4	1.4	31.6	80.6	79.0	**84.2**	–	<u>81.2</u>
Battery Try	$M(n)$	16	20	28	<u>48</u>	35	41	45	**64**
Blocks Ranking Try	$M(n)$	6	26	24	87	81	**100**	23	<u>97</u>
Cover Blocks	$M(n)$	0	0	19	56	<u>97</u>	**98**	84	96
Press Button	$M(n)$	0	0	8	34	3	<u>87</u>	74	**92**
Average	$M(n)$	5.5	11.5	19.8	56.3	54.0	<u>81.5</u>	56.5	**87.3**
Total Average	–	10.4	5.9	26.3	69.8	67.8	<u>83.0</u>	–	**83.9**

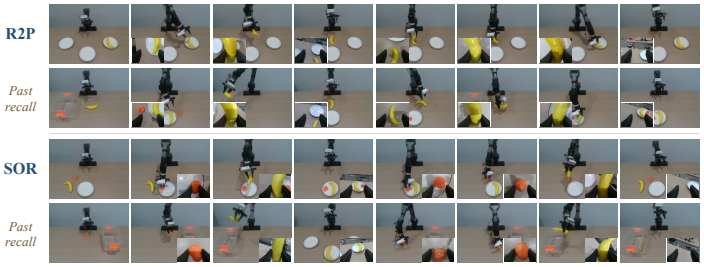
above 97.8% (Table 1). It attains joint-best Object performance and the best Goal score; on Long, its 97.8% lies between LingBot-VA's 98.5% and Fast-WAM's 95.2%. Across these suites, our design supports standard and extended manipulation; the component comparisons below examine the complementary roles of recollection and repertoire in supporting this performance.

A pronounced advantage in repeated-interaction tasks. On RMBench, WA*R*M averages 83.9% versus MemoryWAM's 83.0% across nine tasks; its clearer gain is on $M(n)$, at 87.3% versus 81.5% (Table 2). The largest task gain is on Battery Try, at 64% versus 41%; Press Button improves from 87% to 92%. These tasks require successive insertion attempts or tracking digit-specified press counts (Chen et al., 2026, Appendix A). These repeated behaviors make action reuse conditional on episode progress: an insertion may warrant another attempt, whereas a button press may already satisfy the required count. Our advantage in this setting supports recollection as the episode-specific evidence that determines when to repeat or revise a familiar interaction.

4.3 Real-World Evaluation of Experience Reuse (Q1)

We evaluate four physical tasks: Banana-to-Box (B2B) and Duck-to-Box (D2B) test repertoire reuse across layouts, while Return-to-Plate (R2P) and Same-Order-Retrieval (SOR) test recollection of episode-specific destinations and order using familiar objects. Figure 3a pairs retrieved cross-task events with head- and wrist-view execution: R2P returns the banana to its original plate, and SOR retrieves objects in their placement order, showing familiar manipulation guided by current-episode evidence. Figure 3b reports 17, 16, 15, and 15 successes out of 20, with R2P and SOR gaining seven and six successes over the strongest baseline. These gains highlight recollection's value when reuse

(a) Real-world Execution Sequences



(b) Successful trials / 20

Task	$\pi_{0.5}$	Fast-WAM	WA*R*M
B2B	14	13	**17**
D2B	13	12	**16**
R2P	8	7	**15**
SOR	9	8	**15**

Figure 3: **Physical evaluation of retrospective reuse.** (a) R2P and SOR execution with retrieved observations from events in other tasks. (b) Autonomous successes out of 20 for all four tasks; Appendix C.3 presents the full seven-condition visualization and protocol.

Table 3: **Component comparisons (%).** The upper block varies store access; the lower block tests source initialization and explicit comparison. Rankings span both blocks; Appendix B.2 defines the inputs and generation pathways retained or changed in each of the six variants.

Variant	LIBERO		RMBench	
	Avg. ↑	Long ↑	$M(1)$ ↑	$M(n)$ ↑
Neither event store	97.8	95.2	2.4	14.7
Situated recollection only	98.3	96.6	54.6	59.2
Sensorimotor repertoire only	98.1	96.2	48.9	51.8
Full WA*R*M	**98.7**	**97.8**	**81.2**	**87.3**
Gaussian source (context only)	98.0	97.2	76.8	79.5
Without explicit comparison κ_i	98.2	96.4	77.6	80.0

depends on episode history. Together, these results demonstrate that our retrospection framework carries familiar behavior across physical tasks and layouts while retaining episode-specific control. Appendix C.3 provides seven deployment sequences and further protocol details.

4.4 Contributions of Recollection and Action Reuse (Q2)

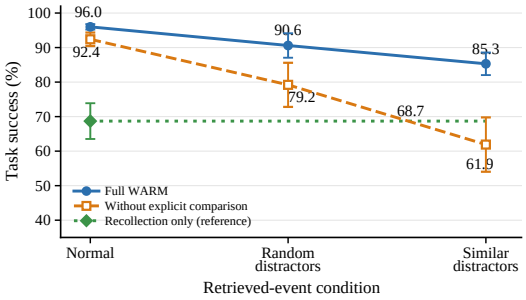
Current evidence and past actions contribute together. Combining the stores gives the strongest results in Table 3. On $M(1)$ and $M(n)$, full WA*R*M exceeds recollection alone by 26.6 and 28.1 points, and repertoire alone by 32.3 and 35.5 points. LIBERO-Long shows the same ordering: 95.2% with neither store, 96.6% with recollection, 96.2% with repertoire, and 97.8% with both. The larger gains on RMBench support the central division of roles: recollection supplies evidence about this episode, while the repertoire supplies actions whose usefulness depends on that evidence.

Assessing experience and shaping generation both matter. The lower block separates experience-informed initialization from explicit consequence comparison. With gated candidate conditioning retained, full WA*R*M gains 4.4 points on $M(1)$ and 7.8 on $M(n)$ over Gaussian initialization. Explicit compatibility κ_i adds 3.6 and 7.3 points over a control that retains action adaptation and effect prediction. Together, these comparisons connect consequence assessment to our two generative pathways: the source supplies action structure, while conditioning guides its refinement. Appendix A.3 decomposes the source mean shift and noise contraction analytically.

4.5 Robust Selection Under Misleading Experience (Q3)

Testing familiar actions in the wrong context. On Put Back Block and Press Button, we replace half the retrieved candidates with random events from the task/control space or visually similar events with incompatible stage or effect evidence. Each donor retains its action, effect, and metadata, preserving a complete interaction record for assessment. Comparing full WA*R*M with the no-

(a) Experience distraction



(b) Drop from normal retrieval

Drop (percentage points) ↓	Random	Similar
Full WA*R*M	**5.4**	**10.7**
Without explicit comparison	13.2	30.5
Difference in degradation ↑	7.8	19.8

Bold/underlining rank the two model drops. The final row reports their difference.

Figure 4: **Robustness to inapplicable experience.** (a) Full WA*R*M and the no-comparison variant under three retrieval conditions on Put Back Block and Press Button. Curves average the two tasks across five evaluation runs (100 resets per task/run); bars show run-level SD. The dotted line is the recollection-only reference. (b) Drops from the same means. Appendix B.2 specifies the intervention.

comparison variant under these perturbations tests the value of explicit consequence compatibility when retrieved experience is only partly useful for the current task (Figure 4).

Explicit comparison preserves useful reuse. With similar-event replacement, WA*R*M retains 85.3% SR versus 61.9% without explicit comparison, and stays 16.6 points above the recollection-only reference. Its degradation is 10.7 points versus 30.5, a 19.8-point advantage compared with 7.8 points for random replacement. Both tasks exhibit this pattern (Appendix C.1). The larger advantage against similar distractors supports the purpose of consequence-aligned recall: current requirements help distinguish applicable behavior from experience that merely looks familiar.

4.6 Candidate-Level Assessment of Action Consequences (Q4)

Our candidate-level protocol evaluates historical effects E_i, the query-only requirement r_t, and predictions $\widehat{E}_i$ against outcomes from executing each adapted proposal μ_i for $H = 32$ steps from a shared restored state (Appendix C.2.1). All three references share branch outcomes and independent applicability labels. MSE measures effect error; ordering agreement tests whether predicted scores preserve the ranking of observed candidate effects under r_t. Applicable choice assesses whether the selected proposal performs the task-appropriate next interaction. In Table 4, prediction reduces MSE from 0.055 to 0.032, raises ordering agreement from 0.414 to 0.642, and improves applicable choice from 17.6% to 19.0% relative to historical effects. The query-only forecast assigns the same score to every candidate; action-conditioned prediction distinguishes their consequences. Together, these metrics link predicted consequences to the task relevance of recalled actions.

Table 4: **Candidate-effect assessment.** Rows share states, proposals, and independent labels. Query-only ordering and ties are analytical; Appendix C.2.1 defines the metrics and shared branch protocol.

Effect reference	MSE ↓	Ordering ↑	Tie rate	Applicable choice ↑
Historical E_i	0.055	0.414	0.03%	17.6%
Query-only r_t	0.043	$1/2$ (ties)	100% (ties)	16.3%
Predicted $\widehat{E}_i$	**0.032**	**0.642**	0.00%	**19.0%**

5 Conclusion

We introduce WA*R*M for event-grounded retrospection, using current-episode evidence to guide selective reuse of past behavior. From current context and situated recollection, we infer a candidate-independent transition requirement and compare it with the predicted effects of adapted repertoire actions. Consequence-gated source transport then jointly regulates stochastic initialization and candidate-derived conditioning, making recalled behavior a revisable prior for action generation. Experiments on LIBERO and RMBench support the complementary roles of recollection and repertoire; complete-event distraction tests show how explicit consequence comparison preserves useful reuse. These findings support a design principle for embodied control: current evidence should govern both the selection of past behavior and its influence on action generation.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting, language editing, LaTeX organization, and consistency checks. We take full responsibility for the final technical content, citations, analyses, and reported results.

ETHICS STATEMENT

This work studies experience reuse for robotic manipulation using simulation benchmarks and real-world robot manipulation data. The data are used to learn and evaluate robot policies, rather than to study human participants. WA*R*M's consequence-based assessment and shared gating regulate the influence of recalled action proposals on action generation; they do not replace system-level safety measures. Physical robot operation requires deployment-specific safety assessment, appropriate human supervision, emergency-stop capability, and safeguards suited to the platform and workspace.

REPRODUCIBILITY STATEMENT

Section 3 specifies WA*R*M's architecture, training objectives, and causal execution procedure. Appendix A details event construction, recollection updates, semantic interfaces, auxiliary supervision, and effective gating, and provides the assumptions and proofs for the theoretical results. Appendix B describes benchmark interfaces, data-partition and retrieval-exclusion rules, baseline evaluation and references, component and event-replacement protocols, and metric aggregation. Appendix C.1 reports task-level experience-distraction results.

REFERENCES

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. *arXiv preprint arXiv:2512.13030*, 2025.

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 *Robotics: Science and Systems XXI*, 2025. doi: 10.15607/RSS.2025.XXI.010.

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.

Tianxing Chen, Yuran Wang, Mingleyang Li, Yan Qin, Hao Shi, Zixuan Li, Yifan Hu, Yingsheng Zhang, Kaixuan Wang, Yue Chen, Hongcheng Wang, Renjing Xu, Ruihai Wu, Yao Mu, Yaodong Yang, Hao Dong, and Ping Luo. RMBench: Memory-dependent robotic manipulation benchmark with insights into policy design. *arXiv preprint arXiv:2603.01229*, 2026.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems XIX*, 2023. doi: 10.15607/RSS.2023.XIX.026.

Confucius. *The Analects of Confucius (from the Chinese Classics)*. Project Gutenberg, 2002. URL `https://www.gutenberg.org/ebooks/3330`. Translated by James Legge. EBook No. 3330. Book II, Chapter XI.

Yinpei Dai, Hongze Fu, Jayjun Lee, Yuejiang Liu, Haoran Zhang, Jianing Yang, Chelsea Finn, Nima Fazeli, and Joyce Chai. RoboMME: Benchmarking and understanding memory for robotic generalist policies. *arXiv preprint arXiv:2603.04639*, 2026.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172, 2023. doi: 10.52202/075280-0403.

Zhiyuan Guan, Jianshu Hu, Han Fang, Yunpeng Jiang, Yize Huang, Shujia Li, Xiao Li, and Yutong Ban. Dssp: Diffusion state space policy with full-history encoding. *arXiv preprint arXiv:2605.14598*, 2026.

Xinying Guo, Chenxi Jiang, Hyun Bin Kim, Yuhang Han, Ying Sun, Yang Xiao, and Jianfei Yang. Chameleon: Control-indexed prospective memory for visuomotor manipulation. *arXiv preprint arXiv:2603.24576*, 2026.

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*, pp. 24328–24346. PMLR, 2025.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4651–4664. PMLR, 2021.

Jindou Jia, Gen Li, Xiangyu Chen, Tuo An, Yuxuan Hu, Jingliang Li, Xinying Guo, and Jianfei Yang. Action-to-action flow matching. *arXiv preprint arXiv:2602.07322*, 2026.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998. doi: 10.1016/S0004-3702(98)00023-X.

Sinjae Kang, Chanyoung Kim, Kaixin Wang, Li Zhao, and Kimin Lee. WarmPrior: Straightening flow-matching policies with temporal priors. *arXiv preprint arXiv:2605.13959*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Huashuo Lei, Wenxuan Song, Huarui Zhang, Jieyuan Pei, Jiayi Chen, Haodong Yan, Han Zhao, Pengxiang Ding, Zhipeng Zhang, Lida Huang, Donglin Wang, Yan Wang, and Haoang Li. RoboMemArena: A comprehensive and challenging robotic memory benchmark. *arXiv preprint arXiv:2605.10921*, 2026.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, 2023a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023b.

Xiangkai Ma, Yue Ma, Junjie Wang, Sheng Xu, Mingyang Li, Han Zhang, Yuzheng Zhuang, Wenzhong Li, and Zhihao Yuan. Decoupling intention from trajectory: A representational deduction framework for world action models. *arXiv preprint arXiv:2608.06994*, 2026.

Bruno Machado, Alexandre Chapin, Emmanuel Dellandrea, and Liming Chen. Flowing with purpose: Latent action guided flow matching policies for robotic manipulation. *arXiv preprint arXiv:2606.23420*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Jeongeun Park, Juhan Park, Taekyung Kim, Sungjoon Choi, Dongyoon Han, and Sangdoo Yun. Retrieve, don't retrain: Extending vision language action models to new tasks at test time. *arXiv preprint arXiv:2606.15631*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. MemoryVLA: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025.

Ajay Sridhar, Jennifer Pan, Satvik Sharma, and Chelsea Finn. Memer: Scaling up memory for robot control via experience retrieval, 2025. URL https://arxiv.org/abs/2510.20328.

Xiaoquan Sun, Ruijian Zhang, Chen Cao, Yihan Sun, Jiahui Chen, Zetian Xu, Bo Chen, Haijier Chen, Zhen Yang, Jiarun Zhu, Yijun Hong, Jingzhe Xu, Jingrui Pang, Mingqi Yuan, and Jiayu Chen. HiMem-WAM: Hierarchical memory-gated world action models for robotic manipulation. *arXiv preprint arXiv:2606.10363*, 2026.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *International Conference on Learning Representations*, 2025.

Marcel Torne, Karl Pertsch, Homer Walke, Kyle Vedder, Suraj Nair, Brian Ichter, Allen Z. Ren, Haohuan Wang, Jiaming Tang, Kyle Stachowicz, Karan Dhabalia, Michael Equi, Quan Vuong, Jost Tobias Springenberg, Sergey Levine, Chelsea Finn, and Danny Driess. MEM: Multi-scale embodied memory for vision language action models. *arXiv preprint arXiv:2603.03596*, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Kai Wang, Zhaopeng Gu, Yixiang Chen, Yuan Xu, Qisen Ma, Jiabing Yang, Zhaowen Li, Yan Huang, Liang Wang, and Peng Su. DIM-WAM: World-action modeling with diverse historical event memory. *arXiv preprint arXiv:2606.27677*, 2026.

Ganlin Yang, Zhangzheng Tu, Yuqiang Yang, Sitong Mao, Junyi Dong, Tianxing Chen, Jiaqi Peng, Jing Xiong, Jiafei Cao, Jifeng Dai, et al. Eventvla: Event-driven visual evidence memory for long-horizon vision-language-action policies. *arXiv preprint arXiv:2606.20092*, 2026a.

Sizhe Yang, Juncheng Mu, Tianming Wei, Chenhao Lu, Xiaofan Li, Linning Xu, Zhengrong Xue, Zhecheng Yuan, Dahua Lin, Jiangmiao Pang, and Huazhe Xu. MemoryWAM: Efficient world action modeling with persistent memory. *arXiv preprint arXiv:2606.20562*, 2026b.

Yi Yang, Zhihong Liu, Siqi Kou, Yiyang Chen, Yanzhe Hu, Jianbo Zhou, Boyuan Zhao, Zhijie Wei, Xiao Xia, Xueqi Li, Pengfei Liu, and Zhijie Deng. World-language-action model for unified world modeling, language reasoning, and action synthesis. *arXiv preprint arXiv:2606.05979*, 2026c.

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

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.

Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, Jiaming Zhou, Yishu Shen, Yudong Jin, Fangyi Xu, Shuailei Ma, Jiaqi Liao, Guanxing Lu, Zifan Shi, Yongkun Wen, Yujie Zhao, Weixuan Tang, Xinyang Wang, Chaojian Li, Jiapeng Zhu, Ka Leong Cheng, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Native video-action pretraining for generalizable robot control. *arXiv preprint arXiv:2607.08639*, 2026.

Sizhe Zhao, Haozhe Xie, Weiyu Zhao, Chenchu Zhang, Huan Wang, Chenyang Wang, Qinglin Liu, and Shengping Zhang. Memory as plans: World-action modeling with memory-grounded planning. *arXiv preprint arXiv:2609.11561*, 2026.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Dai Zhou, Jiexi Yan, Tong Li, Yuxuan Wang, and Cheng Deng. Remember smarter: Visual history compressor and hyperbolic experience space for robotic memory. *arXiv preprint arXiv:2608.15269*, 2026a.

Yijing Zhou, Qiwei Liang, Sitong Zhuang, Jiaxi Li, Xianpeng Wang, Boyang Cai, Yunyang Mo, and Renjing Xu. Action-effect memory pretraining for robot manipulation. *arXiv preprint arXiv:2606.12499*, 2026b.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025.

A Method Details and Mathematical Analysis

We first specify the interfaces that connect recollection, consequence assessment, and action generation, then establish how the shared gate controls recalled experience. Training objectives and causal record construction complete the method specification; Appendix A.6 indexes the core notation.

A.1 Semantic Interfaces and Information Flow

A common space for requirements and effects. The fixed descriptor map $\mathcal{P}_{\text{eff}}$ defines a shared space for historical effects E_i, requirements r_t, and predictions $\widehat{E}_i$. Frozen DINOv2 features Φ (Oquab et al., 2024) are pooled to 2×2 cells of width d_s, with pre-state S_{ij}^- and change ΔS_{ij}:

$$a_{ij} = \|\Delta S_{ij}\|_2^2 / d_s, \quad \omega_{ij} = \frac{a_{ij} + \epsilon_p}{\sum_{k=1}^{4} a_{ik} + 4\epsilon_p}, \quad E_i = \sum_{j=1}^{4} \omega_{ij} (\Delta S_{ij} + 0.05 S_{ij}^-). \tag{13}$$

The stabilizer $\epsilon_p > 0$ handles negligible changes, and $0.05 S_{ij}^-$ preserves starting context. The same map gives the query-demonstration target $E_t^\star$, describing endpoint changes at the encoder's semantic resolution.

Connect current features to stored evidence. We read the same current frame at Video DiT layers 9 and 19, obtaining token streams $U_t^{(1)}, U_t^{(2)}$ of width 3,072; these are two network depths, not two camera views. Learned projections P_1, P_2 and mixing logits b_1, b_2 fuse them:

$$W_t = \omega_1 P_1 U_t^{(1)} + \omega_2 P_2 U_t^{(2)}, \qquad (\omega_1, \omega_2) = \mathrm{softmax}(b_1, b_2), \tag{14}$$

Four learned Perceiver-style queries (Jaegle et al., 2021) compress W_t into $\widehat{S}_t \in \mathbb{R}^{4\times d_s}$, aligned to frozen $\Phi(o_t)$ during training. This bridge connects the instruction- and state-conditioned video representation to stored evidence through the interfaces in Table 5.

Table 5: **Interfaces for consequence-aligned reuse.** Demonstrated actions and futures provide training targets; deployment reads use the inputs listed here.

Computation	Direct inputs	Output and role
Requirement read	$W_t, \widehat{S}_t, q_t, \ell, \mathcal{M}_t$; retrieved groups masked	R_t, r_t: assessment reference
Action adaptation	A_i, c_t, c_i^0, τ_i; current context and R_t	A_i^w, μ_i: aligned and adapted actions
Effect prediction	$\mathrm{stats}(\mu_i), E_i, \mathrm{pool}(W_t)$	$\widehat{E}_i$: proposed effect
Ranking and gating	Candidate/context features, timing, history; gate statistics in Eq. (17)	$\ell_i^{\mathrm{rank}}, \alpha_t, g_t$: selection and influence
Selected-event read	Requirement inputs plus selected pre-state/change tokens	R_t^{pred}: transition context
Action-context readout	Adaptation states, stored change, predicted effect	C_i: projected action/effect context
Action DiT	$x_\sigma, \sigma, \mathcal{C}_t$	v_θ: velocity generating A_t

Two reads, one Transition Adapter. The shared G_{tr} uses eight learned queries, two cross-attention/feed-forward blocks (Vaswani et al., 2017), type embeddings, validity masks, and the native robot-state token. Masking retrieved groups gives $R_t \in \mathbb{R}^{8\times d_r}$ of adapter width d_r; P_{req} projects their mean to the candidate-independent reference $r_t \in \mathbb{R}^{d_s}$. A second read adds the selected event's pre-state/change tokens to produce R_t^{pred}. We gate its increment over R_t by Eq. (9) and apply affine P_{ctx} to reach decoder width.

From an action proposal to its predicted effect. The adapter f_A combines aligned actions A_i^w, position and gripper timing τ_i, current context, and R_t to produce bounded μ_i in Eq. (4). Effect prediction uses mean command, terminal command, and mean absolute temporal variation:

$$\mathrm{stats}(A) = \left[\frac{1}{H}\sum_{h=0}^{H-1} A_h,\ A_{H-1},\ \frac{1}{H-1}\sum_{h=0}^{H-2} |A_{h+1} - A_h|\right] \in \mathbb{R}^{3d_a}. \tag{15}$$

For $H = 1$, variation is zero. World pooling applies a learned projection and validity-masked mean to W_t. Together with E_i and $\mathrm{stats}(\mu_i)$, it gives $\widehat{E}_i$ for the proposal preceding flow refinement. Task context enters through W_t and $R_t \to \mu_i \to \widehat{E}_i$.

Action-context tokens combine adaptation states, stored changes, and predicted effects. Their affine projection to decoder width is included in C_i, so gating scales the entire contribution.

Compare direction and magnitude. For effect vectors p, q above the cosine norm floor, Eq. (6) decomposes as

$$1 - \kappa(p,q) = \frac{1}{2}\left\| \frac{p}{\|p\|_2} - \frac{q}{\|q\|_2} \right\|_2^2 + \lambda_m \left| \log \frac{\|p\|_2 + \epsilon}{\|q\|_2 + \epsilon} \right|. \tag{16}$$

The two terms measure angular mismatch and relative effect magnitude. We use $\lambda_m = 0.25$ and $\epsilon = 10^{-6}$; safe cosine normalizes with denominator floor ϵ, assigning zero cosine to a zero vector. Our action summary compresses temporal order, and the descriptor compresses physical change; Appendix C.2.1 tests whether this resolution preserves useful candidate ordering against independently executed outcomes.

A.2 Recall Selection and Effective Gating

Retrieve, align, and select. For a unit-normalized frozen-DINO CLS vector $v_t \in \mathbb{R}^{768}$ and a one-hot task vector $e(\ell) \in \mathbb{R}^J$, the retrieval key is $k_t = 2^{-1/2}[v_t; e(\ell)]$; stored keys use the same construction. We apply cosine top-K search to these keys, with $J = 40, 9, 4$ for LIBERO, RMBench, and the physical task set, respectively. The task block contributes similarity without excluding cross-task events. Alignment $\mathcal{W}$ preserves relative EEF commands. For absolute joint commands, c_t and c_i^0 denote current and stored starts in the same joint coordinates; their normalized difference shifts arm targets while preserving gripper commands. When such matched start-state coordinates are unavailable, $\mathcal{W}$ leaves the stored command unchanged and the action adapter supplies the learned correction. The adapter then applies the bounded residual in Eq. (4). The reranker reads current/candidate context, adapted-action statistics, observed effects, timing, and phase/ordinal features to produce ℓ_i^{rank}.

Preserve useful event continuity. Let the previously selected event come from demonstration d_0 at time s_0, and let $\Delta_i = s_i - s_0$ for candidates from that demonstration. The recorded-order prior is

$$b_{t,i} = \begin{cases} 0, & \text{no active event or a different demonstration,} \\ -\lambda_{\text{reuse}}, & \text{demo}(i) = d_0,\ \Delta_i \le 0, \\ \lambda_{\text{forward}} \max(0, 1 - \Delta_i / F), & \text{demo}(i) = d_0,\ \Delta_i > 0, \end{cases}$$

with nonnegative reuse penalty λ_{reuse}, forward weight λ_{forward}, and window $F > 0$. Same-or-earlier events are also source-ineligible; successor metadata may supply a continuation candidate. This cursor tracks selected-event order, while recollection tracks actual interaction. It resets each episode; advancement, null-step expiry, and continuation/truncation settings belong to the run specification in Table 8.

Measure adaptation and repetition. For selected residual $\delta = \mu_{i^\star} - A^w_{i^\star}$, the smooth RMS deformation is

$$d = \sqrt{(Hd_a)^{-1}\|\delta\|_F^2 + \epsilon_d^2} - \epsilon_d, \qquad \epsilon_d = 10^{-6}.$$

For L executed-action summary slots, validity flags m_j, and stored repetition coordinates $r_j \in [0, 1]$, we use

$$\zeta_t = \frac{\sum_{j=1}^{L} a_j m_j r_j}{\max(1, \sum_{j=1}^{L} a_j m_j)}.$$

Recency weights a_j are equally spaced from 0.5 to 1 (0.5 for one slot); empty history gives zero. Each summary concatenates mean arm commands, terminal/net arm displacement, and terminal gripper values, with gripper entries zeroed in the arm blocks. Relative commands use their sum for net displacement; absolute targets use offsets from the observed prefix-start state in both arm blocks.

We match a new executed-prefix summary to retained summaries by maximum cosine similarity, breaking ties by relative distance. A repetition flag requires both similarity and distance thresholds to pass; it is zero without earlier history. Observed progress clears it when frozen-feature change reaches the motion threshold, or both frozen-feature and VAE changes exceed the stationary threshold. Reference defaults are 0.97, 0.20, 0.08, and 0.03; Table 8 records evaluated settings.

Turn selection evidence into shared influence. For valid candidates, set $P_i = \text{softmax}_i(\ell_{t,i}^{\text{sel}} / \tau_{\text{sel}})$ with $\tau_{\text{sel}} > 0$. Let $P = P_{i^\star}$ and ΔP be its margin over the strongest competitor. Normalized entropy is $\mathcal{H} = -\sum_i P_i \log P_i / \log |\mathcal{I}_t|$ for multiple candidates; a single candidate has zero entropy and an absent competitor of probability zero. The learned gate combines selection statistics, compatibility, support coordinate n, deformation, and repetition:

$$\alpha_t = \text{sigmoid}\, f_g([P, \Delta P, 1 - \mathcal{H}, \kappa_{i^\star}, \log(1 + n), d, \zeta_t])\,. \tag{17}$$

For each selected valid event, $n = 1$ (and $n = 0$ in padded slots), so $\log(1 + n)$ is a fixed validity coordinate, not an estimated support count. Equation (35) trains α_t as an imitation-utility score; eligibility and repetition determine its effective influence:

$$g_t = \chi_t\, \mathbf{1}[\zeta_t < \zeta_{\max}]\, \alpha_t \exp(-\lambda_\zeta \zeta_t). \tag{18}$$

The indicator χ_t encodes candidate availability and eligibility: absent candidates, forced null, and forced negatives set it to zero; deployment also applies the raw-gate threshold and temporal eligibility. The coefficient $\lambda_\zeta \geq 0$ sets repetition decay and $\zeta_{\max} \in [0, 1]$ sets its cutoff. This same g_t controls the stochastic source and candidate conditioning.

Empty retrieval bypasses argmax, zeros candidate payloads, and sets $g_t = 0$. The candidate-null interface retains the requirement context $P_{\text{ctx}}(R_t)$, native context, visual K/V, and fixed conditioning masks, with zero action-context slots. This common interface supports the exact withdrawal property below.

A.3 Mathematical Properties of Selective Generative Reuse

We establish exact withdrawal and bounded partial influence, then characterize the source and its flow-matching objective. For withdrawal and source geometry, fix the current query and proposal $\mu = \mu_{i^\star}$, vectorize normalized actions in $\mathbb{R}^D$ with $D = Hd_a$, and write $\beta = 1 - \sigma_{\min}$ and $c(g) = 1 - \beta g$. At prescribed gate $g \in [0, 1]$, the source is $z_g = g\mu + c(g)\varepsilon_t$, where $\varepsilon_t \sim \mathcal{N}(0, I_D)$.

A.3.1 Joint withdrawal and bounded action influence

Proof of Proposition 1. Fix factual inputs, recollection, parameters, masks, solver schedule, and sampling randomness for one replan. The candidate-independent read preserves R_t, native context, and visual K/V under candidate replacement. At $g = 0$, Eqs. (8) and (9) give $z_0 = \varepsilon_t$, $R_t^{\text{act}} = R_t$, and $C_t^{\text{act}} = 0$. Every decoder input is identical; induction over solver steps therefore gives identical iterates and final actions. □

Partial admission within a fixed proposal branch. Hold $\mu, R_t, R_t^{\text{pred}}, C_{i^\star}$, factual context, and masks fixed, and vary g with a shared noise draw. For the linear part $P_{\text{ctx}}^{\text{lin}}$ of the affine context projection, define

$$h_t = \left[\text{vec}\Big(P_{\text{ctx}}^{\text{lin}}(R_t^{\text{pred}} - R_t)\Big)\,; \text{vec}(C_{i^\star})\right]. \tag{19}$$

Gating the entire projected candidate contribution gives $\mathcal{C}_g - \mathcal{C}_0 = gh_t$ in the varying coordinates. Assume continuous-flow solutions exist on the unit interval and the following bound holds uniformly over their trajectories and contexts for finite $L_x, L_C \geq 0$:

$$\|v_\theta(x, \sigma; \mathcal{C}) - v_\theta(y, \sigma; \mathcal{C}')\| \leq L_x\|x - y\| + L_C\|\mathcal{C} - \mathcal{C}'\|, \tag{20}$$

With $\psi(L) = (e^L - 1)/L$ for $L > 0$ and $\psi(0) = 1$, we obtain

$$\|A_t(g) - A_t(0)\| \leq g\left[e^{L_x}\|\mu - \beta\varepsilon_t\| + L_C\psi(L_x)\|h_t\|\right] \equiv gK_t. \tag{21}$$

Here $A_t(g)$ is the continuous normalized output before discontinuous command processing, and K_t is independent of g on the fixed branch.

Proof. Set $s = 1-\sigma$, $Y_g(s) = x_{1-s}(g)$, and $D(s) = \|Y_g(s) - Y_0(s)\|$. Then $\mathrm{d}Y_g/\mathrm{d}s = -v_\theta(Y_g, 1 - s; \mathcal{C}_g)$, and $D(0) = g\|\mu - \beta\varepsilon_t\|$. Equation (20) yields

$$D(s) \leq D(0) + \int_0^s [L_x D(u) + gL_C\|h_t\|]\,\mathrm{d}u.$$

Gronwall's inequality gives $D(s) \leq e^{L_x s}D(0) + gL_C\|h_t\|\int_0^s e^{L_x(s-u)}\mathrm{d}u$; at $s = 1$ this is Eq. (21), including $L_x = 0$. □

For a shared Euler grid with positive steps Δ_k summing to one, the recurrence $D_{k+1} \leq (1 + L_x\Delta_k)D_k + gL_C\Delta_k\|h_t\|$ gives the same bound using $\prod_k(1 + L_x\Delta_k) \leq e^{L_x}$, under the same regularity on the iterates. Thus the shared gate controls both source and conditioning through a linearly shrinking influence envelope. The comparison holds within fixed candidate, mask, and eligibility choices, before physical execution; output differences inside the envelope may vary nonmonotonically. Appendix C.2.4 checks the corresponding zero-gate withdrawal in the implementation.

A.3.2 Geometry of the experience-informed source

Writing $\varepsilon = \varepsilon_t$, the source law conditioned on a proposal and gate fixed before noise is drawn is

$$\nu_g = \text{Law}(z_g \mid \mu, g) = \mathcal{N}(g\mu, c(g)^2 I_D), \qquad c(g) \geq \sigma_{\min} > 0. \tag{22}$$

The positive floor keeps the source nondegenerate at full acceptance; across queries, these Gaussian sources can form a mixture. For a fixed demonstrated action $A^\star$, the zero-mean cross term gives

$$\mathbb{E}_\varepsilon \|z_g - A^\star\|_F^2 = \|g\mu - A^\star\|_F^2 + c(g)^2 D, \tag{23}$$

Writing this expected distance as $J(g)$, its change from a Gaussian start is

$$J(g) - J(0) = g^2(\|\mu\|_F^2 + \beta^2 D) - 2g(\langle \mu, A^\star \rangle_F + \beta D). \tag{24}$$

The expression separates the proposal shift from noise contraction, clarifying the two contributions of the shared source. For the candidate-null law $\nu_0 = \mathcal{N}(0, I_D)$, the quadratic Wasserstein distance is

$$W_2(\nu_g, \nu_0) = g\sqrt{\|\mu\|_F^2 + \beta^2 D}. \tag{25}$$

The common-noise coupling attains squared cost $g^2\|\mu\|_F^2 + (c(g) - 1)^2 D$; Cauchy–Schwarz bounds every coupling's centered cross moment by $Dc(g)$, establishing optimality. Hence source displacement scales linearly with admission, while the learned flow determines the resulting action.

A.3.3 Conditional flow matching with experience-dependent sources

The following specialization of conditional flow matching (Lipman et al., 2023; Liu et al., 2023b) accommodates the dependence between a demonstrated action and its retrospective source, including training corruptions.

Proposition 2 (Flow matching with experience-dependent sources) *Fix the retrospective maps and consider full-valid action chunks. Let $(A^\star, Z, \mathcal{C})$ have any joint distribution with finite second moments, where $\mathcal{C}$ is precisely the decoder-visible condition. Define $X_\sigma = (1-\sigma)A^\star + \sigma Z$ and $V = Z - A^\star$. Assume $\mathbb{E}[w(\sigma)\|V\|^2] < \infty$. For independently sampled σ with positive density and $w(\sigma) > 0$, the unrestricted minimizer of the weighted squared flow loss is, almost everywhere,*

$$u_\sigma(x; c) = \mathbb{E}[V \mid X_\sigma = x, \mathcal{C} = c]. \tag{26}$$

The conditional path law $p_\sigma(\cdot \mid c)$ satisfies the continuity equation with field u_σ in the weak sense. If its ODE and continuity equation are well posed backward on each $[\delta, 1]$, $\delta > 0$, sampling from $p_1(\cdot \mid c)$ and integrating the field recovers $p_\delta(\cdot \mid c)$, with weak endpoint limit $p_0(\cdot \mid c)$.

Proof. For fixed σ, c, conditional expectation gives

$$\mathbb{E}[\|v(X_\sigma, \sigma; c) - V\|^2 \mid c] = \mathbb{E}[\|v(X_\sigma, \sigma; c) - u_\sigma(X_\sigma; c)\|^2 \mid c] + \mathbb{E}[\|V - u_\sigma(X_\sigma; c)\|^2 \mid c]. \tag{27}$$

The cross term vanishes because $\mathbb{E}[V - u_\sigma(X_\sigma; c) \mid X_\sigma, c] = 0$. Positive scheduler weights and fixed action-dimension normalization preserve this pointwise minimizer. For a smooth compactly supported test function φ,

$$\frac{\mathrm{d}}{\mathrm{d}\sigma}\mathbb{E}[\varphi(X_\sigma) \mid c] = \mathbb{E}[\nabla\varphi(X_\sigma)^\top V \mid c] = \mathbb{E}[\nabla\varphi(X_\sigma)^\top u_\sigma(X_\sigma; c) \mid c]. \tag{28}$$

Finite second moments and bounded $\nabla\varphi$ justify differentiation under the expectation. This is the weak continuity equation $\partial_\sigma p_\sigma + \nabla \cdot (p_\sigma u_\sigma) = 0$. Uniqueness on $[\delta, 1]$ identifies the ODE law initialized from p_1 with p_σ; mean-square convergence $X_\delta \to A^\star$ gives the weak endpoint limit. □

This result connects experience-conditioned initialization to the population objective under the decoder's actual information interface. Transport assumes an optimal field and matching source sampling; deployment eligibility, finite data, joint updates, model approximation, and discretization are assessed through the empirical policy evaluations.

A.4 Auxiliary Supervision and Gradient Flow

We train each retrospective decision with an explicit target: observed effects supervise semantic prediction, demonstration agreement supervises ranking and adaptation, and selected-proposal agreement supervises raw utility.

Masks and distances. Let $m_{t,h} \in \{0,1\}$ mark valid action steps and $f_t \in \{0,1\}$ mark availability of the future observation. For queries with at least one valid step, define

$$d_A(U,V) = \frac{\sum_{h=0}^{H-1} m_{t,h}\|U_h - V_h\|_2^2}{d_a \sum_{h=0}^{H-1} m_{t,h}}, \qquad d_E(p,q) = \frac{1}{d_s}\|p-q\|_2^2. \tag{29}$$

Candidate sums and softmaxes use $\mathcal{I}_t$. We write $\langle\cdot\rangle_{\mathcal{S}}$ for a query-set mean, with empty means equal to zero. Sets $\mathcal{A}, \mathcal{F}, \mathcal{V}$ contain action-valid queries, those with future targets, and those with valid candidates, respectively; $\mathcal{F}, \mathcal{V} \subseteq \mathcal{A}$. Set $\mathcal{N}$ contains forced hard negatives. Missing futures retain available action supervision.

Requirement and semantic bridge. For prediction p and detached target y, our semantic loss combines direction and log-magnitude alignment:

$$\ell_{\text{sem}}(p,y) = 1 - \cos(p,y) + \lambda_m \,\text{SmoothL1}(\log(1+\|p\|_2), \log(1+\|y\|_2)) \,. \tag{30}$$

Here SmoothL1 denotes smooth absolute error. We set $\mathcal{L}_{\text{req}} = \langle \ell_{\text{sem}}(r_t, \text{sg}(E_t^\star))\rangle_{\mathcal{F}}$ and average $1 - \cos(\widehat{S}_{t,j}, \text{sg}(\Phi(o_t)_j))$ over queries and four cells for $\mathcal{L}_{\text{bridge}}$. The frozen encoder supplies stable observed targets. Training uses a smooth robust penalty on $\log(1+\|\cdot\|_2)$; selection uses the absolute difference of stabilized log magnitudes in Eq. (6). Both compare effect scale, and λ_m weights that comparison.

Ranking and action adaptation. For $t \in \mathcal{V}$, aligned stored actions and historical effects define detached utility weights:

$$u_i = -d_A(A_i^w, A_t^\star) - \lambda_E f_t d_E(E_i, E_t^\star), \qquad y_i = \text{softmax}_{i\in\mathcal{I}_t}(u_i/\tau_u). \tag{31}$$

Here $\tau_u > 0$ sets concentration and $\lambda_E \geq 0$ balances effect agreement, omitted when $f_t = 0$. These labels depend on deterministic alignment and recorded evidence, remaining fixed as the learned proposal and prediction change. We optimize

$$\begin{aligned} \mathcal{L}_{\text{rank}} &= \left\langle D_{\text{KL}}\big(\text{sg}(y) \,\|\, \text{softmax}(\ell^{\text{rank}})\big)\right\rangle_{\mathcal{V}}, \\ \mathcal{L}_{\text{adapt}} &= \left\langle \sum_{i\in\mathcal{I}_t} \text{sg}(y_i) d_A(\mu_i, A_t^\star) \right\rangle_{\mathcal{V}\setminus\mathcal{N}}. \end{aligned} \tag{32}$$

The Kullback–Leibler loss D_{KL} trains relative preference among valid candidates; the weighted action loss trains bounded adaptation, excluding forced negatives.

Factual effect supervision and candidate regularization. An observed transition $(o_t, A_t^\star, o_{t+H})$ trains the shared head with its historical-effect input set to zero:

$$\mathcal{L}_{\text{fact}} = \langle \ell_{\text{sem}}(f_E(\text{stats}(A_t^\star), 0, \text{pool}(W_t)), \text{sg}(E_t^\star))\rangle_{\mathcal{F}} \,. \tag{33}$$

For $\mathcal{R} = (\mathcal{V} \cap \mathcal{F}) \setminus \mathcal{N}$, we also train the candidate-input path with detached utility weights:

$$\mathcal{L}_{\text{cand}} = \left\langle \sum_{i\in\mathcal{I}_t} \text{sg}(y_i) \ell_{\text{sem}}(\widehat{E}_i, \text{sg}(E_t^\star)) \right\rangle_{\mathcal{R}}. \tag{34}$$

We use $\mathcal{L}_{\text{effect}} = (\mathcal{L}_{\text{fact}} + \mathcal{L}_{\text{cand}})/2$ when $\mathcal{R}$ is nonempty, and $\mathcal{L}_{\text{fact}}$ otherwise. The candidate term is a shared-query regularizer: predicting $E_t^\star$ for every candidate minimizes it alone for nonzero targets above the norm floor. Appendix C.2.1 therefore evaluates candidate sensitivity against independently executed effects, comparing $\widehat{E}_i$ with r_t and E_i.

Utility targets and retrieval corruptions. Selected-proposal agreement defines the detached soft target for raw utility:

$$y_t^g = \text{sg}\left[\exp\left(-\frac{d_A(\mu_{i^\star}, A_t^\star) + \lambda_g f_t d_E(\widehat{E}_{i^\star}, E_t^\star)}{\tau_g}\right)\right], \qquad \mathcal{L}_{\text{gate}} = \langle \text{BCE}(\alpha_t, y_t^g)\rangle_{\mathcal{A}} \,. \tag{35}$$

Here $\tau_g > 0$ sets sharpness and $\lambda_g \geq 0$ weights effect agreement, omitted when $f_t = 0$. Null and forced-negative targets are zero; binary cross-entropy (BCE) includes all action-valid rows and is

evaluated from logits for stability. The target measures imitation utility and evolves as the proposal and effect models are updated.

Training mixes normal retrieval, candidate dropout, forced null, and hard negatives selected by demonstrated action/effect disagreement. Forced negatives train ranking and rejection with $g_t = 0$ on both generation inputs. They are excluded from adaptation and candidate-effect losses, while their factual transition continues to supervise requirement and effect prediction.

Flow objectives and gradient routes. For action flow, we draw $u \sim \mathcal{U}(0,1)$ and set $\sigma = 5u/(1 + 4u)$. The scheduler weight $w(\sigma)$ is proportional to $\exp[-2(\sigma - 1/2)^2]$ after subtracting its minimum on the 1,000-point training grid and normalizing the grid mean to one. At inference, Euler integration uses the same shifted grid at $u_k = 1 - k/N$ for $k = 0, \ldots, N$, with $N = 10$ except $N = 8$ for Observe and Pick Up (Table 8). Action-flow training applies this weight with the valid-action mask; a separate video-only pass computes $\mathcal{L}^v_{\mathrm{FM}}$ from observed-video targets with its corresponding schedule. The action pass obtains visual K/V from current-frame processing. We detach z_t in both x_σ and $v^\star$, and detach g_t in conditioning while preserving gradients through the token representations. Thus action-flow gradients train permitted conditioning paths, auxiliary losses train retrospective decisions, and gate supervision also reaches its feature inputs. Retrieval and argmax choose discrete indices; selected continuous readouts remain trainable.

We update Action DiT, retrospective modules, the native state projection, and rank-16 bottleneck adapters at video layers 9 and 19. The VAE, text encoder, Φ, and remaining video-backbone parameters stay fixed. Table 8 specifies the evaluated loss weights and training settings.

A.5 Event Construction and Causal Recollection

Build intact action–effect exemplars. Each repertoire event contains H recorded commands and their boundary observations from a training demonstration. Window starts combine regular anchors, a tail-aligned window, interaction-change peaks, and independently retained gripper transitions. We normalize RMS changes in actions, robot state, and available visual features by their median and median absolute deviation (MAD), average the signals, and select local maxima with nearby-peak suppression. Only complete windows at the recorded control rate are retained, storing $(S_i^-, A_i, \Delta S_i)$, start state, timing, and provenance separately from retrieval keys. The bank stays fixed at evaluation; offline retrieval excludes the query episode and content-identical visual features.

Retain observed interaction. Recollection starts with the initial observed anchor. After each executed prefix, we form four change components $c_{t,k}$: any gripper transition, commanded-direction change, frozen-feature change, and VAE-latent change. Feature changes use $d_{\mathrm{rel}}(u,v) = \min\{1, \|u - v\|_2/(\|u\|_2 + \|v\|_2 + \epsilon_0)\}$ with $\epsilon_0 = 10^{-8}$ and zero for two negligible inputs. Direction change is $(1 - \cos)/2$ between consecutive terminal/net arm displacements; the first prefix averages adjacent nonzero command-vector changes. Absolute-joint summaries use start-relative offsets, and relative controls use displacements. With nonnegative weights of positive total weight, $s_t^{\mathrm{write}} = \sum_k w_k c_{t,k} / \sum_k w_k$.

We write an event when this score exceeds $\mathrm{median}(\mathcal{Q}) + 1.4826\lambda_{\mathrm{MAD}}\,\mathrm{MAD}(\mathcal{Q})$, clipped to configured bounds; warm-up uses the lower bound. The bounded queue $\mathcal{Q}$ contains earlier scores, and receives the current score only after the decision. This order makes offline prefix reconstruction and online retention use identical available evidence.

Compress and read bounded recollection. New events have unit mass and post-observation representatives. We keep the initial anchor separately and protect the latest event. At capacity, we merge the most similar eligible adjacent interior pair by cosine similarity of flattened features, choosing the earliest pair on ties. For representatives f_a, f_b of masses m_a, m_b, the merged feature is $(m_a f_a + m_b f_b)/(m_a + m_b)$ with combined mass, older pre-state, and newer post-state. Repertoire actions remain intact. Readout uses role/relative-age embeddings for initial, historical, latest, and executed-action records, plus separate validity bits for each gripper's opening and closing times. Table 8 records capacities, weights, and thresholds.

Read, commit, and execute causally. At replan t, we read earlier records and already-executed commands alongside current o_t, q_t. After inference, we commit the current observation with its

preceding commands; newly generated commands enter history after execution, and the next observation supplies their feedback. We reuse frozen retrieval features and reset recollection each episode. Identical observed-and-executed prefixes therefore produce identical records under fixed settings: induction preserves the write evidence, earlier-score queue, and eligible merge pairs at every update.

A.6 Core Notation and Module Index

Table 6 prioritizes the main method's recurring symbols and essential module interfaces. Candidate quantities refer to the current query t unless indicated; hats mark learned representations, stars mark demonstration targets, and $i^\star$ is the selected index. Local proof and diagnostic variables are defined where used; numerical settings appear in Table 8.

Table 6: **Core notation by computational role.** Definition references connect the main method to its detailed specification.

Symbol or module	Meaning and role	Definition
Interaction, experience, and actions		
t, o_t, q_t, ℓ	Replan time, RGB observation, robot state, instruction.	§3
$\mathcal{M}_t, \mathcal{B}$	Current-episode recollection; fixed cross-episode repertoire.	§3.1
H, h_{exec}, d_a	Predicted horizon, executed prefix, per-command action width.	§3
$e_i, S_i^-, \Delta S_i$	Stored event, pre-state features, observed feature change.	Eq. (1)
A_i, A_i^w, μ_i	Stored, aligned, and adapted action chunks in $\mathbb{R}^{H\times d_a}$.	Eq. (4)
$A_t, A_t^\star$	Generated action chunk and demonstrated target.	§3.4, 3.5
Semantic representations and learned interfaces		
$\Phi, d_s, \mathcal{P}_{\text{eff}}$	Frozen visual encoder, semantic width, fixed effect descriptor map.	Eq. (13)
$W_t, \widehat{S}_t$	Fused world tokens; learned semantic tokens aligned to $\Phi(o_t)$.	App. A.1
$G_{\text{tr}}, P_{\text{req}}, R_t, r_t$	Shared adapter and requirement projection; output tokens and effect-space forecast.	Eq. (2)
$c_t, c_i^0, \mathcal{W}, f_A, \rho, s$	Compatible action starts, alignment, residual adapter, bound, and scales.	Eq. (4)
$f_E, \widehat{E}_i$	Effect head and its prediction for adapted proposal μ_i.	Eq. (5)
$E_i, E_t^\star, E_{t,i}^{\text{obs}}$	Historical effect, query-demonstration target, and independently executed candidate effect.	Eqs. (3),(39)
Selection and shared control of generation		
$\mathcal{I}_t, K, i^\star$	Valid candidates, retrieval count, selected index.	§3.3
$\kappa_i, \lambda_m, \lambda_c$	Effect compatibility, magnitude weight, compatibility weight in selection.	Eqs. (6),(7)
$\ell_i^{\text{rank}}, b_{t,i}, \ell_{t,i}^{\text{sel}}$	Learned relevance, event-order prior, combined selection score.	Eq. (7)
f_g, α_t, g_t	Utility head, raw utility, effective gate shared by both generation inputs.	Eqs. (17),(18)
$z_t, \varepsilon_t, \sigma_{\min}$	Stochastic source, standard Gaussian draw, retained-noise floor.	Eq. (8)
R_t^{pred}, C_i	Selected-event transition read and projected action/effect context.	§3.4
$R_t^{\text{act}}, C_t^{\text{act}}$	Gated transition and action-context inputs.	Eq. (9)
$P_{\text{ctx}}, \mathcal{C}_t$	Affine context projection; full decoder-visible condition, including masks.	App. A.1
$v_\theta, x_\sigma, \sigma$	Action DiT velocity, intermediate action, flow time from source (1) to action (0).	§3.4
Learning and supervision		
d_A, d_E, sg	Normalized action/effect squared distances; stop-gradient.	Eq. (29); §3.4
$\mathcal{L}_{\text{FM}}^a, \mathcal{L}_{\text{FM}}^v$	Action-flow and video-flow objectives.	Eqs. (11),(12)
$\mathcal{L}_j, \lambda_j, \lambda_v$	Six auxiliary losses and auxiliary/video weights in the joint objective.	Eq. (12)

B EXPERIMENTAL DETAILS

We specify the settings, controlled comparisons, and statistical units underlying Section 4. The protocols distinguish the complementary roles of recollection and repertoire, the contribution of generative reuse, and the robustness provided by consequence assessment. Appendix C.2 details candidate assessment and shared withdrawal; Appendix C.3 describes the physical tasks.

B.1 BENCHMARKS, TRAINING, AND EVALUATION SETTINGS

Task requirements. LIBERO evaluates spatial, object, goal, and extended manipulation across four suites (Liu et al., 2023a). RMBench's $M(1)$ tasks test retention of a task-relevant observation, while $M(n)$ tasks require information accumulated over multiple interactions (Chen et al., 2026). Put Back Block uses the block's original pad as its destination after an intermediate move and button press; Observe and Pick Up requires matching a reference object after it is hidden. Blocks Ranking Try uses feedback to search over block orders, and Cover Blocks requires recovering covered objects in a specified color order. These tasks test whether a policy retains the evidence needed to select the next interaction as execution progresses.

Table 7: **Simulation interfaces for WA*R*M.** Horizons and execution prefixes count model action steps. Baseline-specific observation and execution interfaces are described below.

Setting	LIBERO	RMBench
Tasks / rollouts per task per evaluation	40 / 50	9 / 100
RGB views	2	Head and two wrists
Processed resolution	224×224	240×320 per view
Action / proprioception dimensions	7 / 8	14 / 14
Prediction horizon / executed prefix	32 / 10	32 / 4
Retrieval-key width	$768 + 40$	$768 + 9$
Regular event-window stride	32	4

Data and execution. We split data by episode before extracting event windows, build the repertoire from policy-training episodes, and exclude each offline query's source episode under Appendix A.5. The repertoire reindexes those training demonstrations as action–effect events that remain directly accessible at inference. Recollection resets each episode and follows its actual observation/action prefix. LIBERO uses relative end-effector pose commands and a gripper channel; RMBench uses joint-control actions. Alignment operates within each dataset's control coordinates, as specified in Appendix A.2. LIBERO includes 30 settling steps and gripper binarization; RMBench uses RoboTwin simulation (Chen et al., 2025). In the task order of Table 2, RMBench step limits are 250, 700, 500, 1,000, 600, 1,000, 3,500, 1,500, and 1,500. Policy errors, invalid actions, and timeouts count as failures; infrastructure exclusions follow a common recorded rule.

Training and configuration. We initialize the video/action backbone from released Fast-WAM weights (Yuan et al., 2026). Training updates Action DiT, retrospective modules, the native state projection, and rank-16 video adapters at layers 9 and 19; the remaining pretrained components stay fixed (Appendix A.4). Table 8 summarizes the optimizer, memory, objective, and evaluation settings used across the reported configurations.

Baselines. We compare VLA, world–action, and memory-augmented policies.

(I) *VLA policies.* π_0 combines a pretrained vision–language backbone with flow-matching action generation (Black et al., 2025); $\pi_{0.5}$ extends this recipe through heterogeneous co-training and semantic subtask prediction (Physical Intelligence et al., 2025). These baselines provide general-purpose references for visuomotor control and semantic task understanding.

(II) *World–action models.* Motus integrates understanding, video, and action experts with motion-derived latent actions (Bi et al., 2025). LingBot-VA jointly models video and actions autoregressively, retaining historical context through a KV cache and incorporating observed feedback (Li et al., 2026). Fast-WAM learns from video prediction and uses current-frame features to generate actions at

Table 8: **Training and deployment settings.** We list the shared configuration and identify task-specific choices for LIBERO and RMBench.

Component	Configuration
Training and selection	AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$; 5% linear warm-up and cosine decay to 1% of peak learning rate. Peak learning rates are 10^{-4} for LIBERO and 5×10^{-5} for RMBench specialists; weight decay 0.01, BF16, and gradient clipping 1.0. The four-GPU RMBench configuration uses batch $8 \times 4 \times 4 = 128$. Checkpoints are selected on development data and frozen before test evaluation.
Recollection and retrieval	Episode-disjoint retrieval from training-only banks. LIBERO uses six retained events and two action summaries; RMBench task profiles span 6–20 events and 2–8 summaries. The write-threshold range is $[0.08, 0.85]$. Retrieval uses $K = 32$ for most task profiles and $K = 16$ for Observe and Pick Up; the cursor forward window is 256 steps with expiry after three null steps.
Objectives and admission	Ranking weight 0.10; bridge, requirement, effect, gate, and adaptation weights 0.05 each; video/action flow weights 1.0 each. We use $\rho = 0.25$, $\sigma_{\min} = 0.2$, magnitude weight 0.25, and admission threshold 0.15. Normal/drop/null/hard-negative fractions are 0.50/0.125/0.125/0.25 for LIBERO and 0.60/0.10/0.10/0.20 for RMBench. The adapter uses $s_j = 1$ in normalized action coordinates.
Generation and evaluation	Action horizon $H = 32$ with ten Euler steps; the Observe and Pick Up task profile uses eight. Execution prefixes are ten commands for LIBERO and four for RMBench. We fix reset and sampling schedules at seed 3407 and associate each score with its checkpoint and resolved configuration.

inference (Yuan et al., 2026), providing our backbone reference. WLA-0 combines subtask reasoning with dynamics-supervised actions; its optional test-time scaling scores sampled actions by imagined outcomes, while we use standard inference (Yang et al., 2026c).

(III) *Memory-augmented policies.* MemoryWAM combines recent frames, initial anchors, and compressed gist tokens in a persistent visual cache (Yang et al., 2026b). HiMem-WAM learns motion and skill latents, writes compact states at skill boundaries, and gates memory reading (Sun et al., 2026). DIM-WAM retains diverse historical events in multiple banks and supervises task progress to condition video/action denoising (Wang et al., 2026). EventVLA combines visual anchors with predicted event keyframes; its RMBench row uses the visual-anchors-only configuration, preserving initial and recent observations (Yang et al., 2026a). These methods make past task context available to action prediction through structured temporal representations of the ongoing episode.

Baseline evaluation and references. We ran the released implementations where available under their documented evaluation protocols; for methods without a runnable release, we use the cited results. Table 9 identifies the reference versions and benchmark tables, while each method retains its documented observation and execution interface. Our RMBench evaluation trains a separate specialist for each of the nine tasks; task-specific memory and inference profiles are summarized in Table 8.

Table 9: **Baseline references.** Versions and tables identify the benchmark sources; VA denotes EventVLA's visual-anchors setting.

Benchmark	Reference methods	Paper version / table
LIBERO	π_0, $\pi_{0.5}$, Motus, Fast-WAM	Fast-WAM v2, T2 (Yuan et al., 2026)
LIBERO	HiMem-WAM	v1, T2 (Sun et al., 2026)
LIBERO	LingBot-VA; WLA-0	v2, T2; v1, T1 (Li et al., 2026; Yang et al., 2026c)
RMBench	$\pi_{0.5}$, Fast-WAM, MemoryWAM	MemoryWAM v1, T1 (Yang et al., 2026b)
RMBench	HiMem-WAM; DIM-WAM	v1, T1; v1, TII (Sun et al., 2026; Wang et al., 2026)
RMBench	EventVLA (VA); WLA-0	v2, T8; v1, T2 (Yang et al., 2026a;c)

WLA-0 reports the four $M(n)$ tasks on RMBench; unavailable $M(1)$ entries remain blank in Table 2.

Distinctive contribution of WA*R*M. The baselines above use temporal memory, predictive dynamics, and outcome assessment to support control. We make episode evidence govern the reuse of

cross-episode action experience: recollection supplies a candidate-independent requirement against which we assess the predicted effects of adapted repertoire actions. This comparison guides candidate selection and a shared gate over stochastic initialization and candidate-derived conditioning. Our distinctive coupling controls both which past behavior is reused and its influence on generation, with coordinated admission, attenuation, and withdrawal across both paths. Sections 4.4–4.5 test these roles; Appendix B.2 defines the component and misleading-retrieval controls.

B.2 Protocols for Component Contributions and Retrieval Robustness

Separate evidence access from experience use. The comparisons in Table 3 answer two complementary questions. First, how do current-episode evidence and cross-episode actions contribute together? The four store-access variants test recollection, repertoire, and their combination. Second, how does assessed experience improve generation? The Gaussian-source control isolates experience-informed initialization while retaining gated candidate conditioning; the no-comparison control isolates explicit effect–requirement compatibility while retaining effect prediction and adaptation. Table 10 defines the six variants precisely. The component study uses three training seeds, with shared data partitions and task-specific evaluation protocols.

Table 10: **Controls for evidence access and generative reuse.** The first four entries test the two stores; the last two test experience-informed initialization and explicit consequence comparison.

Variant	Change from full WA*R*M
Neither event store	Mask both stores; use Gaussian initialization and no candidate contribution, preserving current-state interfaces.
Recollection only	Disable repertoire retrieval and both candidate contributions, retaining current-episode evidence.
Repertoire only	Mask past observations, executed-action summaries, history-derived queries, repetition evidence, and continuity state.
Full WA*R*M	Retain both sources, explicit comparison, and both candidate-injection paths.
Gaussian source / context only	Set $z_t = \varepsilon_t$; preserve gated candidate conditioning.
Without explicit comparison	Remove κ_i from selection and gate inputs; retain effect features, effect supervision, adaptation, and both paths.

Evaluation-run variation describes reset and sampling variation within a configured policy; it does not count independent training repetitions.

Test consequence assessment under misleading retrieval. On Put Back Block and Press Button, Figure 4 compares full WA*R*M and the no-comparison variant under normal retrieval, random-event replacement, and similar-event replacement. We target replacement of 50% of candidate slots with intact training events. Random donors come from the task/control space; similar donors also match visual pre-state features while differing in stage or effect evidence. Each donor retains its own action, effect, timing, and provenance. This preserves a coherent past interaction while changing its applicability to the current task, directly testing the role of effect–requirement comparison.

Both policies use the same reset schedule, donor-sampling rule, candidate count, and control budget; subsequent observations follow their respective executions. Donor sampling is independent of policy acceptance and final success. When a donor is unavailable, we retain the original slot; the 50% setting therefore specifies the attempted candidate-slot replacement. Privileged stage labels are used only to construct similar-event donors. Candidate assessment against independently executed outcomes is specified in Appendix C.2.1.

B.3 Metrics and Statistical Interpretation

Task-balanced success. Let $Y_{mjrk} \in \{0, 1\}$ indicate success for policy m on trial k of task j in evaluation run r. For N_{jr} trials per task/run and J tasks, define

$$s_{mjr} = \frac{100}{N_{jr}} \sum_{k=1}^{N_{jr}} Y_{mjrk}, \qquad S_{mr} = \frac{1}{J} \sum_{j=1}^{J} s_{mjr}. \tag{36}$$

The nine-task RMBench average weights the two groups by their task counts:

$$\mathrm{SR}_{\mathrm{RMBench}} = \frac{5\mathrm{SR}_{M(1)} + 4\mathrm{SR}_{M(n)}}{9}. \tag{37}$$

We aggregate before rounding. EventVLA's published total is 67.8%; its displayed integer task scores re-average to 67.9%, so Table 2 retains the source total.

Robustness across retrieval conditions. Figure 4 averages the two tasks within each of five evaluation runs and reports the mean and sample SD over runs. The recollection-only normal-condition result provides a common reference for the benefit of retrieval. For full WA*R*M W, the no-comparison variant N, and condition c, define the performance drop d_{mrc} and the advantage in retained robustness Γ_c:

$$d_{mrc} = S_{mr,\mathrm{normal}} - S_{mrc}, \qquad \Gamma_c = \frac{1}{5}\sum_{r=1}^{5}(d_{Nrc} - d_{Wrc}). \tag{38}$$

The reported means give $\Gamma_{\mathrm{random}} = 7.8$ and $\Gamma_{\mathrm{similar}} = 19.8$ percentage points. This contrast measures how much explicit comparison reduces the loss from misleading experience; Appendix C.1 provides the task-level breakdown.

Statistical units. The five runs in Figure 4 and Table 11 vary evaluation resets and sampling; the displayed SD measures variation across those runs. They are distinct from independent training repetitions, and the 20-trial physical comparisons in Figure 3 are reported as observed success counts.

C Additional Results and Mechanism Evaluation

C.1 Task-Level Robustness to Misleading Experience

Table 11 resolves Figure 4 into its two tasks. Similar-event replacement reduces full WA*R*M by 10.8 points on Put Back Block and 10.6 on Press Button, while the no-comparison drops are 30.4 and 30.6 points. The resulting degradation advantages, 19.6 and 20.0 points, show the same robustness pattern across both tasks. Relative to the recollection-only reference, WA*R*M remains ahead by 15.8 and 17.4 points, while the no-comparison variant falls behind by 8.0 and 5.6 points.

Table 11: **Task-level experience-distraction results (%).** Mean ± sample SD across five evaluation runs, with 100 rollouts per task/run. The recollection-only row is the normal-condition reference. Bold/underlined means are best/second-best within each retrieval condition.

Method	Retrieved events	Put Back Block ↑	Press Button ↑
WA*R*M	Normal	**100.0** ± 0.0	**92.0** ± 1.6
	Random distractors	**94.6** ± 3.4	**86.6** ± 4.8
	Similar distractors	**89.2** ± 4.5	**81.4** ± 5.4
Without explicit comparison	Normal	<u>95.8</u> ± 2.2	<u>89.0</u> ± 2.5
	Random distractors	<u>82.6</u> ± 6.5	<u>75.8</u> ± 6.7
	Similar distractors	<u>65.4</u> ± 6.0	<u>58.4</u> ± 10.6
Recollection only	Reference	73.4 ± 4.0	64.0 ± 8.7

The two tasks require recollection of an original destination and repeated-interaction progress, respectively. Across both demands, explicit comparison helps preserve the benefit of recalled experience. We next connect this task-level comparison to predicted consequences and the shared generative interface.

C.2 Candidate Assessment and Selective Admission

Our evidence chain follows the decisions made by WA*R*M: predict a proposal's effect, assess its task relevance, and regulate its influence on generation. We use a shared branch-execution protocol for effect accuracy and task-appropriate choice, followed by an implementation check of the shared-withdrawal boundary.

C.2.1 Effect Accuracy and Task-Appropriate Selection

Shared execution and independent outcomes. Our branch protocol fixes the checkpoint, source episodes, stages, and queries before scoring. For each adapted proposal μ_i, execution starts from restored physical, controller, task-progress, random-generator, and observation/history states. Executing $H = 32$ commands without replanning yields the target

$$E^{\text{obs}}_{t,i} = \mathcal{P}_{\text{eff}}\big(\Phi(o_t), \Phi(o^{\mu_i}_{t+H}) - \Phi(o_t)\big)\,. \tag{39}$$

Effect metrics use branches with complete H-step endpoints; applicability uses independently resolvable task outcomes, including early terminal outcomes when the task criterion is met. We compare the three effect references on the same valid endpoints and labels for each metric. We assign $a_{t,i} \in \{0, 1\}$ by a predefined next-interaction success criterion or blinded outcome review, retaining unresolved labels as unknown. Outcome review provides independent evaluation labels for prediction and admission.

Effect accuracy. We compare $F_{t,i} \in \{E_i, r_t, \widehat{E}_i\}$ with $E^{\text{obs}}_{t,i}$ on shared valid endpoints using the feature-width-normalized MSE d_E in Eq. (29). The three references distinguish a stored historical effect, a candidate-independent requirement, and an action-conditioned prediction in the current world context. We average candidates within query, then queries within source episode, episodes within task, and tasks with equal weights.

Ordering and applicable choice. Let $s^F_{t,i} = \kappa(F_{t,i}, r_t)$ and $s^{\text{obs}}_{t,i} = \kappa(E^{\text{obs}}_{t,i}, r_t)$. The comparable set $\mathcal{P}_t$ contains candidate pairs with observed-score differences exceeding δ_{obs}. With $\text{sgn}_\delta(x) = 0$ for $|x| \le \delta$ and $\text{sign}(x)$ otherwise, ordering agreement is

$$h^F_{t,ij} = \begin{cases} 1/2, & \text{sgn}_{\delta_{\text{pred}}}(s^F_{t,i} - s^F_{t,j}) = 0, \\ 1, & \text{sgn}_{\delta_{\text{pred}}}(s^F_{t,i} - s^F_{t,j}) = \text{sign}(s^{\text{obs}}_{t,i} - s^{\text{obs}}_{t,j}), \\ 0, & \text{otherwise.} \end{cases} \tag{40}$$

We average pairs within query, then queries, episodes, and tasks as above; tie rate is the fraction of comparable pairs with tied predicted scores. Queries without comparable pairs are excluded from ordering agreement. Tolerances are fixed from repeatability checks and development data before scoring.

For the maximum-score tie set $\mathcal{T}^F_t$, applicable choice is $|\mathcal{T}^F_t|^{-1} \sum_{i \in \mathcal{T}^F_t} a_{t,i}$ when all its labels are known, matching uniform tie breaking. On comparable pairs, query-only scores yield $1/2$ ordering agreement and 100% ties analytically; applicable choice equals the candidate-pool applicability rate, which requires outcome labels. These metrics connect requirement-aligned ordering to independently verified task progress.

C.2.2 Candidate-Specific Effects and Full-Selector Quality

This study extends Table 4 and Appendix C.2.1 from effect accuracy to decision quality. The central question is whether $\widehat{E}_i$ distinguishes actions within the same candidate pool and improves the deployed selector, rather than only predicting a query's average future. The expanded evaluation uses a separate cohort and complements, rather than replaces, the main-text diagnostic.

Shared queries, branches, and outcome labels. The cohort comprises 20 held-out source episodes per task on Put Back Block (PBB), Press Button (PB), and Battery Try (BT), with ten prespecified replans per episode and the deployed $K = 32$ candidate slots at each replan. Episodes used for fitting or hyperparameter/checkpoint selection are excluded. The query schedule is fixed before inspecting predictions: five replans come from approach/alignment stages and five from contact/completion stages of the relevant interaction, using simulator task predicates rather than model scores to identify stages. Episodes with fewer available queries are retained with their actual counts; they are not replaced according to outcomes. At each query, we cache the requirement, adapted proposals, rank scores, and cursor state once. We execute each proposal for $H = 32$ commands from the same restored simulator, controller, task-progress, RNG, and factual-history state, following Appendix C.2.1; no Action DiT refinement is applied in this branch diagnostic.

Applicability is a task-progress label, not a threshold on κ_i. For PBB, a branch must advance the correct grasp/return interaction toward the episode's original pad, without placing the block on a wrong pad. For PB, it must advance approach to the required button or register a permitted press without exceeding the remaining count. For BT, it must advance alignment/insertion of the battery currently required by task feedback, without repeating a feedback-rejected attempt. Before evaluation, each sampled stage is assigned one explicit predicate: reaching the task evaluator's contact/alignment region, acquiring the correct object, or completing the required placement/press/insertion. The mapping and geometric tolerances are frozen on development episodes; a branch is positive only if that predicate is reached without an invalid transition. No progress or an incorrect interaction gives $a_{t,i} = 0$; an outcome that cannot be resolved from the simulator state is unknown, not automatically negative. Labels use privileged state only for evaluation, never as policy or selector inputs.

Table 12: **Branch-cohort accounting.** Full endpoints, early task termination, and missing endpoints partition attempted branches. Applicability counts are separate: an early terminal outcome can still have a known label. Coverage and choice accuracy use the same fully labelled query cohort.

Accounting item	PBB	PB	BT
Source episodes / queries	20 / 200	20 / 200	20 / 200
Attempted candidate branches	6,400	6,400	6,400
Complete H-step endpoints	6,304	6,272	6,208
Early task termination	96	128	192
Missing endpoints / unknown labels	0 / 0	0 / 0	0 / 0
Applicable / inapplicable labels	1,152 / 5,248	1,024 / 5,376	896 / 5,504
Common fully labelled queries	200	200	200
Queries with at least two full endpoints	200	200	200
Queries with comparable effect pairs	196	192	186
Queries with an applicable candidate	100	80	70
Applicable-pool coverage (%)	50.0	40.0	35.0

Candidate-specific effect error. For the shared complete-endpoint candidate set $\mathcal{V}_t$ at query t, center each reference $F_{t,i} \in \{E_i, r_t, \widehat{E}_i\}$ and its observed target within that same set:

$$\widetilde{F}_{t,i} = F_{t,i} - \frac{1}{|\mathcal{V}_t|} \sum_{j \in \mathcal{V}_t} F_{t,j}, \quad \widetilde{E}^{\mathrm{obs}}_{t,i} = E^{\mathrm{obs}}_{t,i} - \frac{1}{|\mathcal{V}_t|} \sum_{j \in \mathcal{V}_t} E^{\mathrm{obs}}_{t,j}. \tag{41}$$

The centered error is $|\mathcal{V}_t|^{-1} \sum_{i \in \mathcal{V}_t} d_E(\widetilde{F}_{t,i}, \widetilde{E}^{\mathrm{obs}}_{t,i})$, with d_E from Eq. (29); queries with fewer than two complete endpoints are excluded from this metric and counted separately. Because the centered query-only prediction is zero, this diagnostic tests candidate differences rather than the common query component. Ordinary MSE and ordering agreement retain the definitions in Appendix C.2.1. Ordering tolerances are fixed using repeatability checks on development branches before held-out scoring.

Table 13: **Effect prediction within shared queries.** Centering removes each query's mean effect. Macro averages weight tasks equally; query-only ordering is exactly $1/2$ on comparable pairs by definition, not an empirical gain.

Task	Effect reference	MSE ↓	Centered MSE ↓	Ordering ↑
PBB	Historical E_i	0.050	0.036	0.430
	Query-only r_t	0.039	0.035	0.500
	Predicted $\widehat{E}_i$	0.028	0.021	0.670
PB	Historical E_i	0.058	0.039	0.410
	Query-only r_t	0.043	0.039	0.500
	Predicted $\widehat{E}_i$	0.033	0.024	0.640
BT	Historical E_i	0.063	0.044	0.380
	Query-only r_t	0.048	0.043	0.500
	Predicted $\widehat{E}_i$	0.037	0.029	0.600
Macro	Historical E_i	0.057	0.040	0.407
	Query-only r_t	0.043	0.039	0.500
	Predicted $\widehat{E}_i$	0.033	0.025	0.637

Separate retrieval coverage from selection quality. For selection, we use one common set $\mathcal{Q}$ of queries whose entire valid candidate pool has known applicability labels, including known early-terminal outcomes. Candidates are not removed because their predicted scores are poor or their H-step endpoints are unavailable. Queries with unknown labels remain in the accounting; we additionally report all-query lower/upper bounds by assigning unknown outcomes 0/1. Let $p^s_{t,i}$ be the selection probability for rule s, uniform over its maximum-score ties (absolute score tolerance 10^{-8}), and let $\mathcal{Q}^+ = \{t \in \mathcal{Q} : \max_i a_{t,i} = 1\}$. Define per-query choice accuracy $c^s_t = \sum_i p^s_{t,i} a_{t,i}$, pool coverage $\Pr(t \in \mathcal{Q}^+)$, all-query accuracy $\mathbb{E}[c^s_t]$, and covered-pool accuracy $\mathbb{E}[c^s_t \mid t \in \mathcal{Q}^+]$. We use equal episode weight within each task, assigning equal weight to queries within episode; the conditional metric renormalizes those same query weights on $\mathcal{Q}^+$. Thus all-query accuracy equals coverage times covered-pool accuracy within a task, making retrieval failure distinct from selection failure.

The five rules are uniform random selection; historical compatibility $\kappa(E_i, r_t)$; predicted compatibility $\kappa(\widehat{E}_i, r_t)$; contextual selection $\ell^{\mathrm{rank}}_i + b_{t,i}$; and the full selector in Eq. (7). They share the actual candidate pool, adapted actions, r_t, rank scores, and continuity state; only the scoring rule changes, and no counterfactual selection updates the cached cursor. Removing κ_i here is a *fixed-checkpoint selector diagnostic*, not the separately configured no-comparison policy in Table 3.

Table 14: **Independent applicability of selected actions (%).** Each cell is all-query / covered-pool accuracy. All rules use the same labelled pools; the macro conditional column averages the task-conditional rates, not a pooled ratio.

Selection rule	PBB	PB	BT	Macro
Uniform random	18.0 / 36.0	16.0 / 40.0	14.0 / 40.0	16.0 / 38.7
Historical compatibility	20.0 / 40.0	17.0 / 42.5	15.0 / 42.9	17.3 / 41.8
Predicted compatibility	23.0 / 46.0	20.0 / 50.0	17.0 / 48.6	20.0 / 48.2
$\ell^{\mathrm{rank}} + b$	27.0 / 54.0	23.0 / 57.5	19.0 / 54.3	23.0 / 55.3
Full selector	31.0 / 62.0	28.0 / 70.0	23.0 / 65.7	27.3 / 65.9

Paired uncertainty and interpretation. We resample source episodes within each task 10,000 times, keeping all queries, branches, and scoring rules of each sampled episode together. We report percentile 95% intervals for paired differences and the equal-task mean, treating source episodes rather than candidate pairs as independent units. Table 15 reports the principal contrasts, with task-wise directions visible in the preceding tables.

Table 15: **Paired mechanism contrasts.** Brackets give episode-clustered bootstrap 95% intervals. Centered-MSE contrasts are predicted minus reference; choice contrasts are percentage points.

Contrast	Estimate [95% interval]
Centered MSE: predicted minus query-only	-0.014 $[-0.019, -0.009]$
Centered MSE: predicted minus historical	-0.015 $[-0.020, -0.009]$
All-query choice: predicted minus historical compatibility	$+2.7$ $[0.9, 4.4]$
All-query choice: full selector minus $\ell^{\mathrm{rank}} + b$	$+4.3$ $[2.0, 6.7]$

Candidate discrimination and decision quality. The predicted-versus-query-only centered-error difference is -0.014, while the full selector changes all-query choice accuracy by +4.3 percentage points relative to $\ell^{\mathrm{rank}} + b$ (Table 15). These contrasts distinguish two claims: sensitivity to candidate-specific consequences, and a benefit of that sensitivity for choosing actions. A lower centered error together with a positive paired choice contrast supports both, whereas lower effect error alone did not establish better selection. All-query accuracy (27.3%) is interpreted jointly with pool coverage (41.7%) and covered-pool accuracy (65.9%), separating unavailable experience from selection errors. These are pre-refinement diagnostics, not rollout success rates or evidence that every rejected proposal is useless after refinement.

C.2.3 Adaptive Utility Beyond Deterministic Admission Rules

This study complements Table 3 and Figure 4 by asking whether learned utility regulates recalled experience in practice, beyond the structural zero-gate property tested in Appendix C.2.4.

Record quality and gate values before refinement. We evaluate normal retrieval and the same random/similar complete-event replacements as Appendix B.2 on PBB and PB. For each condition, we prespecify ten replans from each of 20 held-out episodes per task and log the full learned-gate policy's selected proposal, raw α_t, effective g_t, and all rule flags before executing the refined action. We restore each sampled state to execute the selected adapted proposal and label its applicability using the independent task predicates above. No training-negative/null flag is artificially enabled during evaluation. Empty pools have $g_t = 0$ but no selected-candidate applicability label and are recorded separately, without assigning an arbitrary α_t or negative label. Unknown branch labels likewise remain in coverage counts. Empty-pool / unknown-label counts are 0/0 for normal, 0 for random, and 0 for similar retrieval.

To separate learning from rules, define $v_t^{\mathrm{det}} = 1$ when the selected proposal passes availability/provenance, temporal eligibility, and the hard repetition cutoff, *before* the learned threshold $\alpha_{\min}$. Raw-utility summaries and learned rejection rates are conditioned on $v_t^{\mathrm{det}} = 1$; effective-gate summaries include deterministic vetoes for every labelled selected proposal. This prevents a lower mean gate caused by hard rules from being attributed to learned discrimination.

Table 16: **Gate response to selected-proposal quality.** Medians [Q1, Q3] replace a distribution plot. N/N_{det} gives labelled/eligible counts. The final column is $\Pr(\alpha_t < \alpha_{\min} \mid v_t^{\mathrm{det}} = 1, a)$ in percent, with the deployed threshold unchanged. Empty pools and unknown labels are reported separately.

Retrieval	Applicability	N/N_{det}	$\alpha_t \mid v^{\mathrm{det}} = 1$	g_t	Rejected (%)
Normal	Applicable	260 / 252	.72 [.56, .84]	.55 [.38, .68]	4.0
	Inapplicable	140 / 116	.23 [.10, .39]	.04 [.00, .17]	34.5
Random	Applicable	220 / 208	.70 [.54, .82]	.53 [.35, .66]	4.8
	Inapplicable	180 / 140	.19 [.08, .33]	.00 [.00, .13]	46.4
Similar	Applicable	200 / 188	.68 [.52, .81]	.51 [.32, .65]	5.3
	Inapplicable	200 / 148	.14 [.05, .28]	.00 [.00, .08]	54.1

Table 17: **Why candidate influence becomes zero (%).** First-veto attribution uses the displayed left-to-right order, so causes are mutually exclusive. All rates use the full logged-query cohort per condition; the complete overlapping flag vector is retained in the logs. Repetition's continuous attenuation is reflected in g_t, not counted as a hard veto.

Retrieval	Other eligibility	Continuity	Repetition	$\alpha < \alpha_{\min}$	Active $g > 0$
Normal	1.0	4.0	3.0	12.5	79.5
Random	2.0	6.0	5.0	18.8	68.2
Similar	3.0	7.0	6.0	22.5	61.5

The tables condition on the *selected* proposal: a distractor-rich pool does not require a lower gate if selection still finds useful experience. Moreover, α_t is trained as imitation utility, not a calibrated probability that a raw proposal is applicable.

A matched constant-utility control. We train a control that substitutes a development-selected constant α_0 for the learned utility in *both* source and conditioning, while retaining identical deterministic eligibility, continuity, hard/soft repetition, retrieval, selector architecture, losses, and training budget. Candidate availability and eligibility checks, including the comparison to $\alpha_{\min}$, operate on the substituted value. We choose α_0 from $\{0.25, 0.50, 0.75\}$ using short, equally budgeted development-only runs and the equal-task/equal-condition mean SR; ties choose the smaller value. This choice is frozen before full-budget training and held-out evaluation; the selected constant is $\alpha_0 = 0.50$. The control retains the utility head and its auxiliary supervision but disconnects its output from action generation, isolating use of adaptive utility rather than removal of a loss or module. Initialization, data ordering, optimizer schedules, NFE, action horizon/prefix, corruption schedules, and checkpoint-selection rules are matched. Each variant is trained under its own deployed gate rule with three paired training seeds (11, 22, 33). The retrieval/selector procedure is unchanged, but its fitted weights and visited states may differ across trained policies; only the fixed-query diagnostic above holds numerical scores and candidates identical.

Each trained checkpoint is evaluated on 100 paired resets per task/condition, with identical donor-sampling and policy-noise schedules at reset and the existing task step limits. This gives 3,600 evaluation episodes across two variants, three seeds, two tasks, and three conditions, excluding development selection. Subsequent trajectories are allowed to diverge; no shared on-policy candidate sequence is assumed. We report mean $\pm$ sample SD across *training seeds*, after averaging reset outcomes within seed. This control differs from the no-comparison model, and its training-seed variability differs from the five evaluation-run variability in Figure 4.

Table 18: **Learned versus constant utility (%).** Mean $\pm$ sample SD uses three training seeds; macro scores average both tasks within each seed before computing SD. Drop is normal minus similar-distraction SR, computed within seed.

Task	Utility	Normal	Random	Similar	Similar drop $\downarrow$
PBB	Learned	99.3 ± 0.6	94.0 ± 2.0	89.0 ± 1.0	10.3 ± 0.6
	Constant	99.0 ± 1.0	89.7 ± 1.5	78.7 ± 2.1	20.3 ± 1.5
PB	Learned	92.0 ± 1.0	86.3 ± 1.5	81.3 ± 1.2	10.7 ± 1.5
	Constant	91.3 ± 1.5	80.7 ± 2.1	69.7 ± 2.1	21.7 ± 1.2
Macro	Learned	95.7 ± 0.8	90.2 ± 1.4	85.2 ± 1.0	10.5 ± 1.0
	Constant	95.2 ± 1.0	85.2 ± 0.6	74.2 ± 0.6	21.0 ± 0.5

For uncertainty, we retain paired reset outcomes within task and seed and report the three seed-level contrasts separately. A descriptive t interval on paired seed differences uses $\bar{d} \pm t_{0.975,2} s_d/\sqrt{3}$; with only three seeds, this summary depends on a normality assumption and has limited precision. The similar-condition macro contrast is 11.0 points (three seed-level contrasts: 11.5,9.5,12.0; descriptive 95% interval: [7.7,14.3]). The reduction in the normal-to-similar drop is 10.5 points (95% interval: [8.3,12.7]).

Adaptive strength versus structural withdrawal. The matched control distinguishes the value of adaptive strength from the shared rules that disable candidate influence. Comparable normal-condition success together with a smaller distraction-induced drop supports a benefit beyond those unchanged rules. Higher raw utility for applicable proposals among deterministically eligible cases additionally localize that response to learning rather than veto frequency. The closed-loop result is essential: low raw-proposal applicability alone does not prove that refinement cannot make use of the recalled structure. The structural withdrawal guarantee alone does not establish a benefit of learned adaptive strength; that claim depends on the matched closed-loop comparison.

C.2.4 Shared Withdrawal Under Candidate Replacement

We test the candidate-null boundary of Proposition 1 on 50 fixed development prefixes from five Press Button and five Put Back Block episodes. With current inputs, sampling randomness, and conditioning masks fixed, we replace retrieved candidates while forcing both the source and candidate-conditioning gates to zero. The resulting 32×14 action outputs are identical in all 50 comparisons (maximum absolute difference 0; `v8-step300`, BF16, batch one, 20 inference steps). This implementation check confirms that the two candidate pathways withdraw together under the tested fixed-input condition.

C.3 Real-World Tasks and Evaluation Protocol

Banana-to-Box (B2B) and Duck-to-Box (D2B) probe repertoire reuse under changed layouts and poses (Section 4.3). Return-to-Plate (R2P) and Same-Order-Retrieval (SOR) add episode-specific destination and order requirements for the same banana and rubber-duck objects, testing how recollection guides familiar interactions across task goals and scenes. We evaluate 20 autonomous trials per method/task with shared resets and a fixed repertoire; recollection resets between trials.

C.3.1 Platform, Data, and Shared Interface

We use a single-arm AgileX PiPER-series manipulator with a parallel-jaw gripper. Fast-WAM provides a same-family world–action reference, and $\pi_{0.5}$ provides a VLA reference under a shared physical action interface. The comparison shares external (head) and wrist RGB views, proprioception, task instructions, training splits, reset schedules, and execution limits across methods. Table 19 records the data and adaptation configuration.

Recorded demonstrations and preprocessing. The current real-robot dataset contains 25 teleoperated demonstrations, partitioned into 20 training and 5 development episodes. The train/development counts are 4/1 for Banana-to-Box, 4/1 for Duck-to-Box, 8/2 for Return-to-Plate, and 4/1 for Same-Order-Retrieval. The processed catalog contains 12,333 command-bearing frames: 9,976 in training and 2,357 in development. The episode-level training/development split is distinct from the tabletop-variation evaluation in Figure 5. The repertoire contains 1,356 events from the 20 training episodes only. Retrieval uses $K = 32$ and a $768 + 4$-dimensional visual–task key. We account for demonstrations and autonomous policy trials separately.

Table 19: **Physical-data and adaptation configuration.** Data timing describes the demonstration stream; deployment actions use the interface specified below.

Component	Recorded setting
Platform	PiPER-series single arm with parallel-jaw gripper.
Observation	External/head and wrist RGB; declared 640×480 inputs, resized to 224×224 per view and joined into a 224×448 image.
Proprioception	Seven dimensions: TCP position, rotation vector, and gripper width.
Action target	Six absolute joint targets in radians and absolute gripper width in meters.
Temporal interface	Nominal data rate 20 Hz; prediction horizon $H = 32$.
Training / repertoire	20 train / five development demonstrations; 1,356 training-only events; $K = 32$.
Adaptation configuration	BF16; per-device batch one, accumulation one; learning rate 10^{-4} with cosine decay; weight decay 0.01; gradient clipping 1.0.
Randomness / budget	Training seed 42; corruption seed 3407; eight epochs configured.

Consistent state and action semantics. The joint-action conversion retains TCP-based proprioception and republishes joint targets as actions. Training normalization statistics and the event bank are rebuilt for this action representation. Although both vectors have seven dimensions, state and action retain their respective Cartesian and joint-control semantics. For absolute-joint alignment, the control-space starts c_t, c_i^0 refer to joint coordinates; TCP proprioception q_t remains a distinct input to the learned adapters. The 20 Hz value describes demonstration sampling; the joint targets are absolute commands rather than frame-to-frame differences.

The adaptation entries specify the training settings; the nominal data frequency and epoch budget retain their configuration-level meaning.

Shared repertoire and traceable reuse. We build one repertoire from the 20 training episodes across all four tasks. Inspection of its event indices confirms 1,356 training events and excludes the five development episodes. Normalization uses the training partition. Task identity contributes to retrieval similarity (Appendix A.2); source episode and event identifiers preserve action provenance.

Each indexed event contains its source task, episode, action interval, and observation history. The retrieved cross-task observations in Figure 5 make repertoire reuse visible, while the physical tasks distinguish the episode-specific requirement supplied by recollection from the familiar behavior supplied by the repertoire.

C.3.2 Four Tasks and Success Criteria

B2B and D2B: action reuse across layouts. The robot grasps a banana or duck from the table and places it inside the box. Evaluation configurations vary object and box positions and the object's initial pose within the declared workspace. Methods share 20 reset configurations in randomized order. Success requires the correct object to be released fully inside the box within the episode limit, with a stable final placement and no corrective assistance. These tasks test whether familiar repertoire actions remain useful as grasp and placement geometry changes; the simulation component comparisons quantify the contribution of access to the repertoire.

R2P: retain the original destination. A banana initially occupies one of three visually identical plates; the robot moves it to a common table-center staging position and then returns it to its original plate. The instruction refers to the original plate without naming its location. The 20-reset schedule allocates the three starting plates as evenly as possible (7, 7, and 6 trials), independently of layout and method, and randomizes trial order. After staging, the three plates are empty and the banana occupies the same central region; successful completion requires both transfers in order and a stable release on the correct source plate. Trials without a return attempt fail the full-task criterion. Uniform random destination choice succeeds with probability $1/3$ before accounting for the two required transfers.

SOR: retain the order of completed interactions. The same banana and rubber duck used in B2B and D2B are placed one at a time into a common receiving area and then retrieved in the same order. Placement follows trial-varying one-step cues; retrieval refers to the order actually completed. The schedule uses each placement order in ten trials and assigns final object positions independently of order across the evaluation layouts. We design the retrieval phase so that transient placement cues have left the current input and final object arrangement is independent of chronology. Success requires both placements and both retrievals in the recorded order within the episode limit; the first order violation ends sequence success. Logging requested and completed placements separately makes the reference order explicit when an earlier action fails.

Historical cues at the decision point. R2P's original plate and SOR's placement order are established before the corresponding return or retrieval decisions. Staging removes the banana from its initial plate, and SOR's transient placement cues leave the current camera views before retrieval. The robot replans from current observations while retaining each policy's native episode history; the task criteria above score the resulting destination and order decisions.

C.3.3 Outcome Accounting and Reporting

Figure 3b reports autonomous successes out of 20 for each method and task under shared reset schedules. Success follows the task-specific criteria above; timeouts, wrong-object actions, dropped



Figure 5: **Seven physical deployment sequences with retrieved cross-task events.** Within each task pair, the upper row shows left-to-right execution keyframes from the external/head camera with wrist-view insets; the lower *Past recall* row shows observations retrieved from indexed events in other tasks. Original and Black identify the two tabletop appearances. Each pair exposes a familiar interaction recalled under a new task or scene.

objects, order violations, and corrective interventions count as failures. The common denominator makes the four task comparisons directly interpretable, while the R2P destination and SOR order criteria distinguish history-dependent choices from physical execution.

Deployment sequences and retrieved cross-task events. Figure 5 extends the two examples in Figure 3a to seven physical conditions: B2B, D2B, R2P, and SOR on the original tabletop, plus D2B, R2P, and SOR on a black tabletop. These sequences provide qualitative views of the four tasks; Figure 3b reports the 20-trial totals at task level, without separate per-appearance rates. Each task row

follows one deployment from the external/head view, with wrist-view insets; the adjacent *Past recall* row shows observations indexed from retrieved events in other tasks. The pairs connect cross-task interaction histories to the policy's physical execution under the current goal and scene.

Scene variation and repertoire reuse. On the original tabletop, B2B and D2B progress from object pickup to stable placement in the box. The D2B sequence repeats that transfer on the black tabletop, where the background appearance and object–box arrangement change. Across these examples, retrieved events pair banana and duck interactions with different destinations, making reusable grasp-and-transport structure visible across goals. The deployments show WA*R*M carrying that familiar manipulation into the changed scenes while completing the current object-to-box goal.

Recollection in history-dependent tasks. In both R2P scenes, the banana leaves its initial plate, reaches the central staging area, and returns to the same plate. In both SOR scenes, the robot completes the placements and retrieves the objects in their earlier placement order. The object identities remain familiar across all four tasks, while the correct R2P destination and SOR retrieval order depend on the current episode's past. These visible choices connect recollection's episode-specific requirements to the repertoire's familiar manipulation behavior in physical execution.

Together, the seven examples make the two roles of retrospection concrete: selected cross-task events supply reusable action structure, and recollection carries the destination or order established earlier in the current episode. The 20-trial comparison strengthens this visual evidence: WA*R*M succeeds on 15/20 R2P and SOR trials, while each baseline reaches at most 9/20.