MEMALIGN: MEMORY–ACTION ALIGNMENT FOR VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models have achieved remarkable progress in robotic manipulation, particularly when augmented with memory to support history-dependent decision-making under partial observability. However, memory-augmented policies may fail to translate historical evidence into appropriate actions, a gap we refer to as *memory–action misalignment*. In this work, we introduce MEMALIGN, a post-training framework that aligns action generation with historical evidence retained in memory, enabling policies to resolve ambiguous action choices using historical context. Specifically, we first introduce counterfactual supervision to train the policy to select appropriate actions for different histories under the same current observation and instruction. Then, we propose a history-aware low-rank modulation module to dynamically modulate the action network based on historical evidence, allowing different histories to induce different effective action mappings. Finally, we complement this modulation with an optional visual rebinding module that locates remembered targets in the current scene. Their estimated positions then guide the refinement of the generated actions. Extensive experiments on two long-horizon manipulation benchmarks (MIKASA-Robo and RoboMME) demonstrate that MEMALIGN significantly improves the success rate of completing history-specified targets, e.g, raising MemoryVLA's result from 13.0% to 87.0% on MIKASA-Robo tasks. Notably, MEMALIGN exhibits strong generalization capabilities, consistently improving performance when applied to different memory-augmented VLA models.

1 INTRODUCTION

Vision-language-action (VLA) models generate robot actions from visual observations and language instructions, providing a foundation for general-purpose manipulation (Zitkovich et al., 2023; Kim et al., 2025b). By combining pretrained vision-language models with robot demonstration data, models such as RT-2 (Zitkovich et al., 2023), OpenVLA (Kim et al., 2025b), and π_0 (Black et al., 2025b) support instruction following and generalization across manipulation settings. Reliable operation, however, may require evidence beyond the current inputs. Information about occluded objects, past events, and task progress may be unavailable in the current observation, leaving the same scene and instruction consistent with different task states that may require different actions (Kaelbling et al., 1998; Cherepanov et al., 2026a; Dai et al., 2026). In such partial observation settings, interaction history plays a crucial role in selecting appropriate actions. This motivates augmenting VLAs with memory to retain task-relevant information and make it available for subsequent decisions.

Memory-augmented VLAs address this limitation by preserving historical evidence for action generation. Existing methods encode motion history as visual traces (Zheng et al., 2025), propagate historical information through recurrent states (Li et al., 2024b), or retrieve relevant representations from memory banks (Shi et al., 2026b). However, a policy may still act on the wrong target even when its interaction history provides sufficient evidence to identify the correct one. Figure 1 illustrates a task in which a robot must pick up the container hiding the blue cube. Although the current visual observation and task instruction are identical, histories H_1 and H_2 indicate that the cube is hidden in containers A and B, respectively. If a policy selects A under history H_1 but C under history H_2, its behavior changes with history, yet it still chooses the wrong target in the latter case. We refer to the failure to translate historical evidence into the corresponding target-directed behavior as

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.

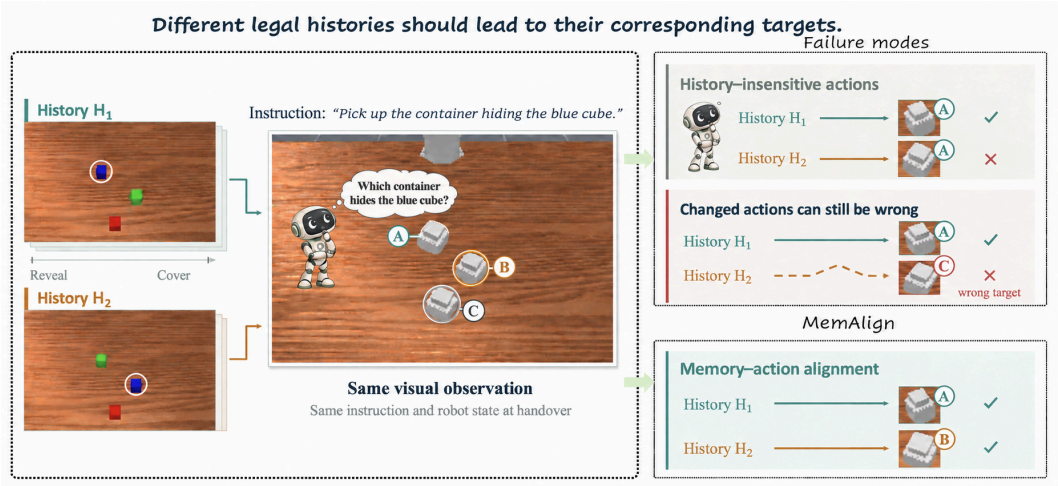


Figure 1: **Memory–action misalignment in memory-augmented VLAs.** Histories H_1 and H_2 indicate that the blue cube is hidden under containers A and B, respectively, while the current visual observation, instruction, and robot state are identical at handover. Selecting A for both histories misses the target change; switching from A to C changes behavior but still fails under H_2. MEMALIGN aims to select A under H_1 and B under H_2. All branches are schematic.

memory–action misalignment. This motivates explicit supervision that connects each history to the actions required to act on its specified target.

To address this gap, we introduce MEMALIGN, a post-training framework that aligns action generation with historical evidence. We first formulate counterfactual supervision using task-consistent alternative histories paired with their corresponding expert actions under the same current observation and instruction. This supervision encourages the policy to resolve ambiguous action choices using historical context. To incorporate this dependence on history into action generation, we introduce a history-aware low-rank modulation module that dynamically modulates the action network using memory features. Its history-dependent adaptation coefficients allow different histories to induce different effective action mappings within a shared policy. We further complement this modulation with an optional visual rebinding module for tasks that require explicit spatial grounding. It matches historical visual cues to the current scene, estimates the corresponding target position, and uses this estimate to refine the generated actions.

We evaluate MEMALIGN on two long-horizon manipulation benchmarks, i.e., MIKASA-Robo and RoboMME. Under a controlled protocol on four MIKASA-Robo tasks, MEMALIGN achieves an average success rate of 87%, substantially outperforming existing VLA baselines. When applied to different memory-augmented VLA models, MEMALIGN further improves their manipulation performance. For instance, on RoboMME, it increases average task success from 8.25% to 10.88% for MemoryVLA, while adapting its counterfactual supervision to MemER improves success from 42.38% to 44.00%. Additionally, MEMALIGN achieves competitive results on LIBERO (Liu et al., 2023) and SIMPLER (Bridge) (Li et al., 2025), demonstrating its applicability to conventional manipulation tasks.

In a nutshell, our contributions are threefold:

- We propose MEMALIGN, a post-training framework that aligns action generation with historical evidence through counterfactual supervision, enabling policies to resolve ambiguous action choices using memory.
- We introduce a history-aware low-rank modulation module that modulates the action network using memory features. An optional visual rebinding module locates remembered targets in the current scene and uses their estimated positions to refine actions.
- Experiments on MIKASA-Robo and RoboMME demonstrate substantial performance gains and show that MEMALIGN effectively uses historical evidence in memory to generate actions directed at the corresponding targets.

2 Related Work

Vision-language-action models (VLAs). VLA models learn visuomotor policies conditioned on natural-language instructions, with discrete token prediction and continuous action generation as two principal formulations. Large-scale imitation learning supports multi-task generalization (Brohan et al., 2023) and cross-embodiment transfer (Open X-Embodiment Collaboration, 2024; Ghosh et al., 2024). PaLM-E incorporates sensory representations into language models, while VIMA conditions manipulation on multimodal prompts (Driess et al., 2023; Jiang et al., 2023). RT-2 and OpenVLA formulate control as autoregressive prediction of discretized actions (Zitkovich et al., 2023; Kim et al., 2025b); FAST introduces compressed action tokenization (Pertsch et al., 2025). Continuous action decoders employ parallel regression (Kim et al., 2025a), diffusion (Chi et al., 2023; Li et al., 2024a), or flow matching (Black et al., 2025b). RT-H introduces language-mediated action hierarchies, while $\pi_{0.5}$ combines semantic subtask prediction with heterogeneous co-training (Belkhale et al., 2024; Black et al., 2025a). Our method introduces counterfactual supervision into a memory-augmented VLA to align historical evidence with target-directed actions.

Memory for robotic manipulation. History-conditioned manipulation addresses ambiguity from occlusion, transient cues, and unobserved task progress. Short observation windows provide local temporal context (Chi et al., 2023); Hiveformer jointly models language, multi-view observations, and interaction history (Guhur et al., 2023), while RoboFlamingo combines pretrained VLM features with a recurrent policy head (Li et al., 2024b). Longer histories can be represented as visual motion traces (Zheng et al., 2025), persistent recurrent states (Li et al., 2026a), or compact moment tokens initialized through time-contrastive learning and aggregated across timesteps (Koo et al., 2026). Explicit memory architectures include MemoryVLA's perceptual-cognitive retrieval mechanism (Shi et al., 2026b) and SAM2Act+'s spatial memory bank and memory attention (Fang et al., 2025). Training objectives form a complementary research line: GR-1 jointly predicts actions and future images (Wu et al., 2024), and Past-Token Prediction reconstructs past actions alongside future action prediction to strengthen long-context modeling (Villasevil et al., 2025). It also selects candidate predictions by their consistency with past actions at inference time. MemoryVLA++ combines memory with future-state imagination (Shi et al., 2026a). MIKASA-Robo and RoboMME provide dedicated evaluations of memory-dependent manipulation (Cherepanov et al., 2026a; Dai et al., 2026), complementing architectural and training advances with tests of temporal reasoning.

Interventions and counterfactual supervision. Imitation policies can exploit predictive correlations that fail during execution. de Haan et al. (2019) identify reliance on causally irrelevant features and use targeted interventions to address this confusion. With observation histories, temporal correlations can induce copycat policies that reproduce previous actions; adversarial representation learning helps suppress this shortcut (Wen et al., 2020). CoDA generates counterfactual transitions by recombining locally independent causal mechanisms for off-policy reinforcement learning (Pitis et al., 2020). Within VLA systems, CF-VLA revises planned meta-actions through outcome reasoning in autonomous driving (Peng et al., 2026), while CoRe uses imagined continuations and physical realignment to recover a frozen manipulation policy at inference time (Zhang et al., 2026). Our work studies target-consistent history interventions under a shared current decision context.

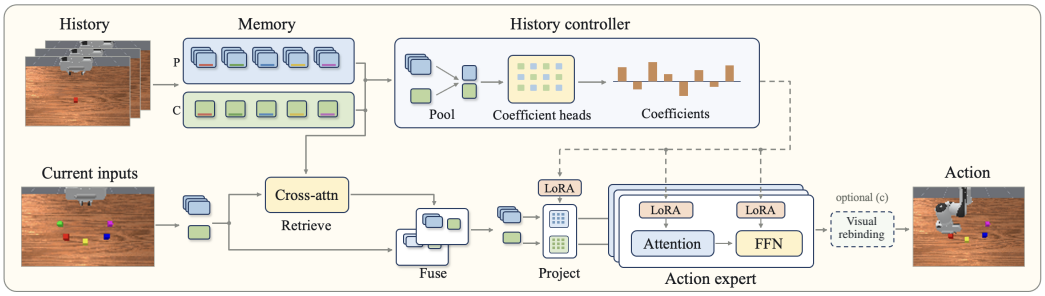
3 Preliminaries

We consider language-conditioned manipulation under partial observability. At decision time t, the current context is $x_t = (o_t, \ell, s_t)$, comprising an RGB observation, an instruction, and proprioception when available. Historical evidence $H_{<t}$ contains only observations strictly preceding t. A stochastic policy generates an action chunk

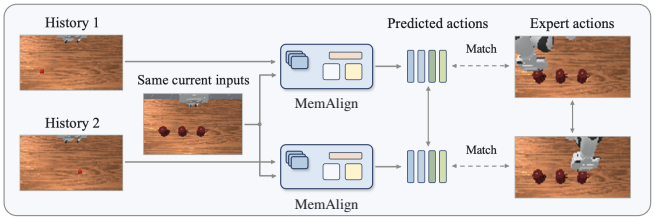
$$\hat{A}_t = \pi_\theta(x_t, H_{<t}; \xi) \in \mathbb{R}^{H \times d_a}, \tag{1}$$

where H is the prediction horizon, d_a the action dimension, and ξ the sampling noise. A prefix is executed before replanning. In memory-dependent decisions, the current context alone does not identify the required target.

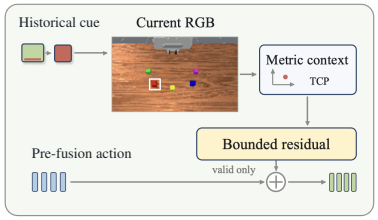
Our implementation builds on MemoryVLA (Shi et al., 2026b), which fuses current features with perceptual and cognitive memory to form a condition c_t for a diffusion action expert. Given a noised



(a) History-aware low-rank modulation



(b) Counterfactual supervision



(c) Visual rebinding (optional)

Figure 2: **Overview of the proposed MEMALIGN. (a)** History-aware low-rank modulation uses a history controller to modulate adapters in the feature projections and action expert. P and C denote perceptual and cognitive memory. **(b)** Counterfactual supervision pairs different histories with their corresponding expert actions under matched current inputs and supervises history-dependent action changes. **(c)** Optional visual rebinding matches historical RGB cues to the current image and uses a valid metric target context to guide bounded action refinement.

expert chunk $A_t^{(k)}$, the denoiser is fitted by

$$\mathcal{L}_{\text{diff}} = \mathbb{E}\Big[\text{MSE}\Big(\epsilon_\theta(A_t^{(k)}, k, c_t), \epsilon\Big)\Big], \tag{2}$$

where the expectation covers expert samples, diffusion indices, and Gaussian noise (Ho et al., 2020; Chi et al., 2023). The forward process is given in Appendix B.2.

4 METHODOLOGY

Figure 2 provides an overview of MEMALIGN. We first introduce counterfactual supervision that aligns action generation with historical evidence under matched current inputs. We then describe a history-aware low-rank modulation module that modulates the action network using memory features. Finally, an optional visual rebinding module locates remembered targets in the current scene and uses their estimated positions to refine actions.

4.1 COUNTERFACTUAL SUPERVISION

Counterfactual supervision teaches the policy how historical evidence should change its actions under the same current context (Figure 2(b)). We define *memory–action alignment* through controlled changes in the behavior specified by history. At a fixed action-handover state, consider a group

$$\mathcal{G}_t = \left\{ \left(H_{<t}^{(i)}, g_i, A_i^\star \right) \right\}_{i=1}^{K}, \tag{3}$$

whose branches share the current context x_t and physical layout. Each legal history $H_{<t}^{(i)}$ specifies a target g_i and is paired with a corresponding expert action $A_i^\star$. An aligned policy should pursue each history's target: replacing $H_{<t}^{(i)}$ with $H_{<t}^{(j)}$ changes the desired behavior from g_i to g_j, rather than corrupting the input while retaining the original target.

Training objectives. We distinguish fitting each history's action from explicitly supervising the behavioral changes associated with different histories. For the MemoryVLA-based implementations, history-matched fitting uses the native noise-prediction loss $\mathcal{L}_{\text{diff}}$ and/or clean-action regression $\mathcal{L}_{\text{act}} = \mathbb{E}[d_{\text{act}}(\hat{A}, A^\star)]$. Each legal history receives its own expert continuation, including when it serves as a donor history.

The Shell v9 continuation instantiates counterfactual supervision as centered action-contrast alignment. For a matched group $\mathcal{G}_t$, predictions $\hat{A}_i$ share sampling noise ξ. Writing $\overline{\hat{A}} = K^{-1}\sum_i \hat{A}_i$ and $\bar{A}^\star = K^{-1}\sum_i A_i^\star$, we use

$$\mathcal{L}_\Delta = \mathbb{E}_{\mathcal{G}_t,\xi}\left[\frac{1}{K}\sum_{i=1}^{K} d_\Delta\left(\hat{A}_i - \overline{\hat{A}}, A_i^\star - \bar{A}^\star\right)\right]. \tag{4}$$

Distances are evaluated in normalized action coordinates over valid entries. Centering emphasizes agreement with the expert's within-group action changes, while matched fitting constrains individual predictions. This term is applied only at decisions with matched current inputs.

More generally, the MemoryVLA adaptation stages use

$$\mathcal{L}_{\text{MemAlign}} = \lambda_{\text{diff}}\mathcal{L}_{\text{diff}} + \lambda_{\text{act}}\mathcal{L}_{\text{act}} + \lambda_{\text{CF}}\mathcal{L}_{\text{CF}}, \tag{5}$$

where the active terms and their weights depend on the training variant. For Shell, $\mathcal{L}_{\text{CF}} = \mathcal{L}_\Delta$ on matched handover batches. The separate Shell component study uses a composite CF objective containing target-choice, action-difference, permutation, and reference-preservation terms. Appendix B specifies the active objectives and action interfaces; the native MemER adaptation is described separately in Appendix D. Counterfactual groups and expert labels are training constructs; deployment uses the available history and current context.

History interventions. We evaluate original, replaced, and erased histories (O/R/E). O and R are scored against their respective targets; E diagnoses behavior without historical evidence and is not assigned a deliberately incorrect action. Current inputs are matched at handover, not throughout the subsequent rollout: observations may diverge once different actions are executed. The intervention protocol is detailed in Appendix C.1.

4.2 History-Aware Low-Rank Modulation

To incorporate historical evidence into the action network, we introduce a history-aware low-rank modulation module (Figure 2(a)). At a controller update time s, the retained perceptual and cognitive memory features $M_{<s}$ are mapped to a history control vector $z_s^{\text{hist}} = C_\phi(M_{<s}, x_s)$. In the RC and Shell implementations, this construction combines history–current feature contrasts, temporal endpoint contrasts, and a Transformer response relative to an all-history-masked reference. The RoboMME implementation uses a simpler pooled-history controller. A coefficient head maps the control vector to layer-specific coefficients $\gamma_s \in \mathbb{R}^r$. At action decision t, γ_t denotes the coefficients from the most recent controller update. For a layer input u, the adapted mapping is

$$y = W_{\text{ref}}u + b + s_h B_h[\gamma_t \odot (A_h u)], \tag{6}$$

where A_h and B_h define a rank-r adaptation subspace and s_h is a scale. The history-independent reference mapping W_{ref} includes any existing static adapter. Histories presented at the same current context can therefore induce different effective mappings within a shared action expert. We apply the modulation at the perceptual/cognitive projections and selected action-DiT transformations.

Coefficients are computed relative to a null-history reference and bounded by an outer $\tanh$. An explicit validity mask sets the additional conditional increment to zero when its historical evidence is absent; other adaptation paths and native memory processing may remain active. Coefficients are reused throughout denoising. Their update frequency follows the evaluation protocol: in the controlled Shell v9 evaluation, the explicit cue context is computed at the step-8 handover and cached for subsequent decisions, while the native memory pathway continues to process the rollout. Appendix A.2 specifies the controller variants, null references, and layer placement.

4.3 VISUAL REBINDING

For tasks requiring explicit spatial grounding, we complement action modulation with an optional visual rebinding module (Figure 2(c)). In the RC implementation, a localizer reads historical RGB cues and the current RGB observation, forms a history-derived appearance reference, and estimates the corresponding target location in the current image. This branch accesses visual history directly; it does not take an explicit target-ID prediction from the LoRA controller as input. The notation $\hat{g}_t$ denotes the entity specified by historical evidence, rather than a separately supervised discrete classifier. Projection into the robot base frame and combination with the current TCP state produce a metric context z_t^{metric} and a validity flag v_t.

Given the action chunk A_t^{pre} generated by the adapted policy, a fuser predicts a residual bounded elementwise by $r_{\max}$:

$$\hat{A}_t = A_t^{\text{pre}} + v_t \, \Delta_\psi\big(A_t^{\text{pre}}, z_t^{\text{metric}}\big) \, . \tag{7}$$

The validity test checks historical evidence availability, usable projection geometry, and the configured confidence criterion. When $v_t = 0$, the fuser returns A_t^{pre} exactly. This fallback preserves the adapted pre-fusion action; the validity test does not guarantee that a geometrically usable estimate identifies the correct object. Appendix A.3 details the RGB localizer, metric projection, and residual parameterization. This metric-fusion branch is used in RC and is absent from Shell v9.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Benchmarks. We conduct our main experiments on two benchmarks for memory-dependent manipulation. MIKASA-Robo (Cherepanov et al., 2026a) tests tabletop manipulation under partial observability, including recalling transient visual cues and locating occluded targets. Our controlled evaluation covers RememberColor3, RememberColor5, RememberColor9, and ShellGameTouch. RoboMME (Dai et al., 2026) assesses temporal, spatial, object, and procedural memory through four task suites: Counting, Permanence, Reference, and Imitation. We evaluate all 16 tasks. Additionally, we evaluate on LIBERO (Liu et al., 2023) and the WidowX/Bridge setting of SIMPLER (Li et al., 2025), which provide complementary evaluations of conventional manipulation performance.

Evaluation Metrics. We report success rates (%). In the controlled MIKASA evaluation, success requires completing only the target specified by the supplied history, termed *exclusive target completion*. Original and Replaced histories are scored against their respective targets; Erased history is used only for diagnosis. Other benchmarks follow their native success criteria. Evaluation protocols, episode counts, and aggregation rules are detailed in Appendix C.

Implementation Details. We implement MEMALIGN on MemoryVLA using rank-8 history-aware modulation while keeping the pretrained backbone frozen. Adaptation modules are optimized with AdamW, and checkpoints are selected on held-out development data. We train separate variants for RememberColor, ShellGameTouch, and RoboMME. The RememberColor5 model is evaluated without further training on RememberColor3 and RememberColor9; visual rebinding is used for RememberColor and selected ShellGameTouch ablation settings. For MemER, we adapt counterfactual supervision to its native subgoal and action interfaces (Appendix D). Implementation and training configurations are provided in Appendices A and B.

5.2 MAIN RESULTS

Results on MIKASA-Robo. Under our controlled protocol, MEMALIGN raises average exclusive target completion from 13.0% for released MemoryVLA to 87.0% (Figure 3(a)). This metric excludes multi-object completion, so the gain reflects more selective execution toward the remembered target. Improvements span cue recall in RememberColor and tracking through occlusion and motion in ShellGameTouch. RememberColor9 remains challenging (60%): most failures complete incorrect or additional targets (Appendix E.1), highlighting residual difficulty with target-directed execution among more candidates. Table 1 provides published baselines under different protocols.

Table 1: **Task completion on MIKASA-Robo (%).** Our results measure exclusive target completion with original histories (50 episodes per task).

Method	RememberColor			ShellGame Touch	Average
	3 objects	5 objects	9 objects		
Octo-small (Ghosh et al., 2024)	45.0	17.0	11.0	46.0	29.75
π_0 (Black et al., 2025b)	35.0	22.0	15.0	33.0	26.25
$\pi_{0.5}$ (Black et al., 2025a)	7.0	12.0	5.0	0.0	6.00
SpatialVLA (Qu et al., 2025)	27.0	17.0	11.0	23.0	19.50
OpenVLA-OFT (Kim et al., 2025a)	59.0	16.0	6.0	47.0	32.00
CronusVLA (Li et al., 2026b)	31.0	13.0	9.0	32.0	21.25
MemoryVLA (Shi et al., 2026b)	44.0	30.0	20.0	88.0	45.50
MemoryVLA++ (Shi et al., 2026a)	50.0	19.0	16.0	97.0	45.50
μVLA (Cherepanov et al., 2026b)	92.0	93.0	41.0	0.0	56.50
SimpleMemVLA (Yin et al., 2026)	71.0	58.0	59.0	99.0	71.75
Gated Memory Policy (Gao et al., 2026)	80.0	61.0	17.0	98.0	64.00
MEMALIGN (Ours)	**100.0**	**98.0**	**60.0**	**90.0**	**87.00**

Results on RoboMME. Across all 16 tasks, MEMALIGN improves average success from 8.25% to 10.88% relative to our locally evaluated MemoryVLA baseline (Appendix E.2). The largest category-level gain occurs in Reference, from 5.5% to 14.0%, driven mainly by VideoPlaceButton and VideoPlaceOrder. This pattern is consistent with improved use of remembered evidence for target-directed execution. Permanence also improves, from 11.0% to 14.5%, whereas Counting decreases from 9.5% to 6.5%. This contrast suggests that the present adaptation is more effective for reference-dependent execution than for counting-dependent behavior. Tables 2 and 3 provide per-task results and within-model comparisons, respectively.

Results on General Manipulation. On LIBERO, MEMALIGN achieves 96.25% aggregate success, compared with 96.75% for MemoryVLA; both achieve 71.88% on SIMPLER (Bridge). Together with the memory-dependent results, these near-baseline aggregates suggest that the main benefit of MEMALIGN lies in using historical evidence to guide behavior. General manipulation performance remains comparable in the evaluated configurations, despite task-level gains and losses (Appendix E.5).

5.3 ABLATION STUDIES AND ANALYSIS

Does Replacing History Redirect the Target? We replace each history with a valid alternative from the same layout that specifies a different target, holding the current observation, instruction, and robot state fixed at handover. Figure 3(b) shows that exclusive completion of the replacement target

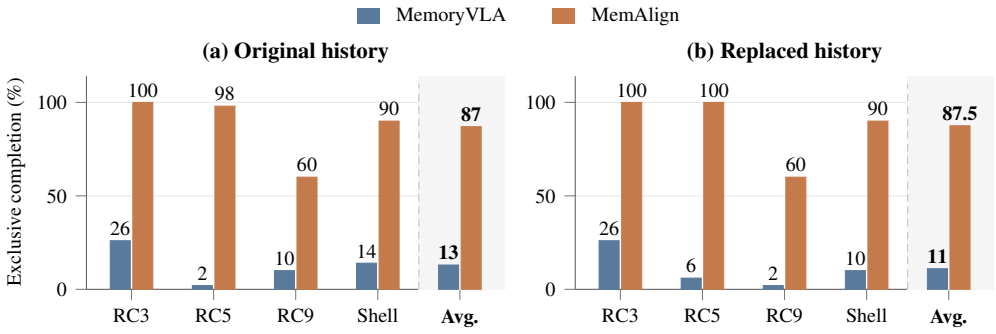


Figure 3: **History-dependent target completion on MIKASA-Robo.** Success requires completing only the target specified by the supplied history (50 episodes per task and condition). Original and replaced histories specify different targets under matched current inputs at handover. RC denotes RememberColor, Shell denotes ShellGameTouch, and Avg. is the mean over four tasks.

Table 2: **Task success on RoboMME (%).** Baseline scores follow Dai et al. (2026); our results use a separate evaluation protocol (Appendix C). Avg. is the mean over all 16 tasks.

Task	OpenVLA -OFT	Diffusion Policy	$\pi_{0.5}$	$\pi_{0.5}$ + past actions	SAM2 Act+	MME-VLA	MemER	Memory VLA	MEMALIGN
Counting									
BinFill	8.00	32.00	30.00	26.67	40.00	39.56	56.67	10.00	**8.00**
PickXtimes	8.00	2.67	42.89	58.33	76.00	87.33	79.33	17.33	**12.00**
SwingXtimes	2.00	6.67	35.56	26.67	25.33	92.00	59.33	1.33	**2.00**
StopCube	2.00	0.00	6.67	4.67	0.00	42.00	0.00	0.00	**4.00**
Permanence									
VideoUnmask	16.00	16.00	20.44	30.67	27.33	32.67	81.33	16.00	**26.00**
ButtonUnmask	0.00	4.00	22.22	23.67	32.00	25.11	72.00	8.00	**20.00**
VideoUnmaskSwap	18.00	4.00	18.67	20.67	18.00	24.44	38.00	4.67	**12.00**
ButtonUnmaskSwap	0.00	0.00	6.67	16.00	26.67	18.22	21.33	4.67	**0.00**
Reference									
PickHighlight	10.00	0.00	11.33	12.33	17.33	22.89	70.67	7.33	**12.00**
VideoRepick	0.00	0.00	0.44	8.67	5.33	30.44	25.33	0.00	**0.00**
VideoPlaceButton	28.00	22.00	31.11	24.00	24.67	60.00	30.00	13.33	**28.00**
VideoPlaceOrder	20.00	5.67	25.78	18.67	20.00	32.00	26.00	4.00	**16.00**
Imitation									
MoveCube	18.00	8.67	26.00	34.00	29.33	77.78	82.67	14.00	**24.00**
InsertPeg	4.00	0.00	1.56	1.00	0.00	7.56	6.67	0.00	**0.00**
PatternLock	6.00	2.00	2.89	4.00	0.00	53.56	16.67	12.00	**8.00**
RouteStick	8.00	10.00	4.67	5.67	0.00	66.67	12.00	1.33	**2.00**
Avg.	9.25	7.11	17.93	19.73	21.37	44.51	42.38	7.12	**10.88**

increases from 11.0% for MemoryVLA to 87.5% for MEMALIGN. Successful episodes complete only the replacement target, directly linking historical evidence to the executed target choice.

Erasing the identifying cue reduces MEMALIGN's original-target completion to 16.0% on average (Appendix E.1), consistent with dependence on historical evidence. Erasure is a diagnostic condition because it leaves the intended target unspecified. Together, replacement and erasure distinguish following the target specified by memory from merely reacting to changes in memory inputs.

Generality across VLA Models. We transfer counterfactual supervision to MemER's native subgoal and action interfaces (Sridhar et al., 2026). Across all 16 RoboMME tasks, with 50 episodes per task, the adapted policy achieves 44.00% success (352/800), 1.62 percentage points above the reported MemER reference score of 42.38% (Table 3). The local MemoryVLA comparison improves from 8.25% to 10.88%, a gain of 2.63 percentage points. These results support the applicability of history-matched adaptation across different memory-to-action interfaces. The MemER comparison uses a published reference under its source protocol.

Table 3: **RoboMME average success (%).** † marks published reference scores.

Model	Baseline	+ MEMALIGN	Δ (pp)
MemoryVLA	8.25	**10.88**	**+2.63**
MemER	42.38†	**44.00**	**+1.62**
MME-VLA	44.51†	**45.62**	**+1.11**

History-Aware Modulation and Counterfactual Supervision. Table 4 holds the backbone, training data, action fitting, and 2,000-update budget fixed. With counterfactual supervision and visual rebinding fixed, replacing static LoRA with history-aware modulation raises original/replaced exclusive completion from 38%/38% to 52%/56%. With history-aware modulation and visual rebinding fixed, adding counterfactual supervision improves these rates from 44%/46% to 52%/56%, and success under *both histories* for the same scene from 12% to 20%. These gains suggest complementary roles: modulation makes the action mapping history-dependent, while supervision specifies its response to historical evidence. This training configuration differs from the main ShellGameTouch result (Appendix E.3).

Generalization across Candidate-Set Sizes. The fixed RC5 model achieves 100% on RC3 and 60% on RC9 under both histories (Figure 3). With the adapter, localizer, and fuser unchanged,

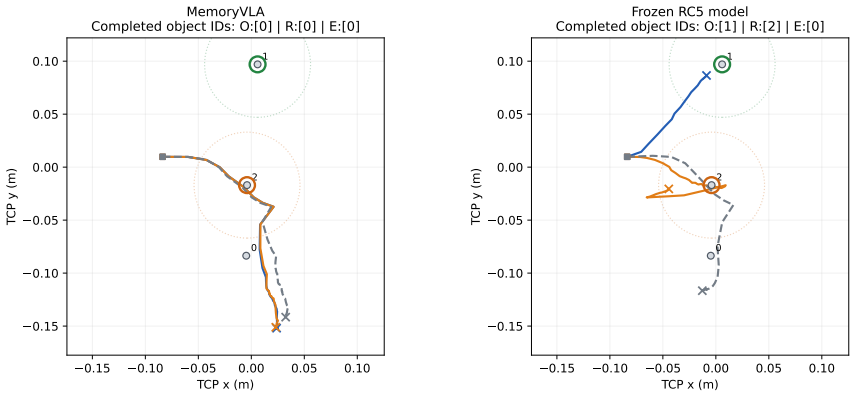


Figure 4: **History-dependent target redirection on RC3.** XY TCP trajectories for MemoryVLA (left) and MEMALIGN (right) under original (O, blue), replaced (R, orange), and erased (E, dashed gray) histories. Squares mark matched handover positions and crosses final positions. Rings project the 0.05 m distance criterion into XY; completion uses three-dimensional distance and the native staticness conditions.

success under replaced histories shows that transfer preserves history-dependent target redirection across candidate-set sizes within RememberColor (Appendix E.4).

Qualitative Target Redirection. From the shared handover state in figure 4, MEMALIGN completes object 1 under the original history and object 2 under the replacement, while MemoryVLA completes object 0 under both. Erasing the cue also leads MEMALIGN to object 0. Figure 3 quantify the consistency of this target redirection.

Table 4: **Component ablation on ShellGameTouch.** All variants use action fitting and 2,000 updates; disabling history-aware modulation gives static LoRA. Results use 50 paired scenes. *Both histories* requires exclusive completion under both original and replaced histories for the same scene. The full configuration is shaded.

Components			Exclusive completion (%) ↑		
History-aware modulation	Counterfactual supervision	Visual rebinding	Original	Replaced	Both histories
—	—	—	22	16	0
—	✓	✓	38	38	4
✓	—	✓	44	46	12
✓	✓	✓	**52**	**56**	**20**

Limitations. Due to hardware constraints, we have not conducted experiments on physical robots, leaving real-world validation to future work. In addition, the inference latency and additional memory usage of MEMALIGN have not been systematically quantified; profiling these costs would help assess its practical deployment requirements.

6 CONCLUSION

We presented MEMALIGN, a post-training framework for addressing memory–action misalignment in memory-augmented VLAs. Counterfactual supervision teaches the policy how its actions should change when different histories specify different targets under the same current context. History-aware low-rank modulation translates memory features into changes in the action network's effective mapping. Optional visual rebinding complements this modulation by locating remembered targets in the current scene and refining actions with their estimated positions. Controlled history interventions on MIKASA-Robo support more reliable execution of history-specified targets, while matched ablations show complementary contributions from modulation and supervision. RoboMME gains and supervision transfer to MemER further support applicability across VLA architectures.

AI USE STATEMENT

AI tools were used to assist with programming, improve the clarity and fluency of the manuscript, and support literature search and background research. They were not used to generate the core research ideas or to conduct experiments in a fully autonomous manner. The authors take full responsibility for the implementation, experimental results, references, and claims presented in this paper.

ETHICS STATEMENT

This work studies history-conditioned robotic control using benchmark datasets and simulated environments. The reported evaluations do not involve real-world robot deployment. Incorrect interpretation of historical evidence or erroneous target selection may lead to unintended actions, which could pose safety risks in physical environments. Our results establish performance under the evaluated protocols rather than safety guarantees for deployment. Any real-world application would require additional validation, task-specific safeguards, and appropriate human oversight.

REPRODUCIBILITY STATEMENT

We document the evaluated model variants and implementation details in Appendix A, and the data construction, training objectives, hyperparameters, and model-selection procedures in Appendix B. Appendix C specifies the history interventions, evaluation settings, sample counts, and metric definitions. The adaptation to MemER is described separately in Appendix D. Appendix E provides detailed outcome counts, benchmark results, and matched ablations, while Appendix F discusses computational resources and limitations of the experimental evidence. Complete trajectory visualizations for the evaluated RememberColor scenes are provided in the supplementary material.

REFERENCES

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quan Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. RT-H: Action hierarchies using language. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.049. URL https://www.roboticsproceedings.org/rss20/p049.html.

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

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025b. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S Ryoo, Grecia Salazar, Pannag R Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan H Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe

Yu, and Brianna Zitkovich. RT-1: Robotics transformer for real-world control at scale. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.025. URL https://www.roboticsproceedings.org/rss19/p025.html.

Egor Cherepanov, Nikita Kachaev, Alexey K. Kovalev, and Aleksandr I. Panov. Memory, benchmark & robots: A benchmark for solving complex tasks with reinforcement learning. In *International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=9cLPurIZMj.

Egor Cherepanov, Nikita Kachaev, Daniil Zelezetsky, Aydar Bulatov, Artem Pshenitsyn, Yuri Kuratov, Alexey Skrynnik, Aleksandr I. Panov, and Alexey K. Kovalev. μVLA: On recurrent memory for partially observable manipulation in VLA models. *arXiv preprint arXiv:2606.12497*, 2026b. URL https://arxiv.org/abs/2606.12497.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://www.roboticsproceedings.org/rss19/p026.html.

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. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.04639.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.nips.cc/paper_files/paper/2019/hash/947018640bf36a2bb609d3557a285329-Abstract.html.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8469–8488. PMLR, 2023. URL https://proceedings.mlr.press/v202/driess23a.html.

Haoquan Fang, Markus Grotz, Wilbert Pumacay, Yi Ru Wang, Dieter Fox, Ranjay Krishna, and Jiafei Duan. SAM2Act: Integrating visual foundation model with a memory architecture for robotic manipulation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15925–15942. PMLR, 2025. URL https://proceedings.mlr.press/v267/fang25c.html.

Yihuai Gao, Jeff Jinyun Liu, Shuang Li, and Shuran Song. Gated Memory Policy: In-Context Memorization and Adaptation, 2026. URL https://arxiv.org/abs/2604.18933.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.090. URL https://www.roboticsproceedings.org/rss20/p090.html.

Pierre-Louis Guhur, Shizhe Chen, Ricardo Garcia Pinel, Makarand Tapaswi, Ivan Laptev, and Cordelia Schmid. Instruction-driven history-aware policies for robotic manipulations. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 175–187. PMLR, 2023. URL https://proceedings.mlr.press/v205/guhur23a.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: Robot manipulation with multimodal prompts. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14975–15022. PMLR, 2023. URL https://proceedings.mlr.press/v202/jiang23b.html.

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.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, 2025a. doi: 10.15607/RSS.2025.XXI.017. URL https://www.roboticsproceedings.org/rss21/p017.html.

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, 2025b. URL https://proceedings.mlr.press/v270/kim25c.html.

Myungkyu Koo, Daewon Choi, Taeyoung Kim, Kyungmin Lee, Changyeon Kim, Younggyo Seo, and Jinwoo Shin. HAMLET: Switch your vision-language-action model into a history-aware policy. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KcJ9U0x6kO.

Hang Li, Fengyi Shen, Dong Chen, Liudi Yang, Xudong Wang, Jinkui Shi, Zhenshan Bing, Ziyuan Liu, and Alois Knoll. ReMem-VLA: Empowering vision-language-action model with memory via dual-level recurrent queries. *arXiv preprint arXiv:2603.12942*, 2026a. URL https://arxiv.org/abs/2603.12942.

Hao Li, Shuai Yang, Yilun Chen, Xinyi Chen, Xiaoda Yang, Yang Tian, Hanqing Wang, Tai Wang, Dahua Lin, Feng Zhao, and Jiangmiao Pang. Towards efficient and robust manipulation via multi-frame vision-language-action modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18388–18396, 2026b. doi: 10.1609/aaai.v40i22.38903. URL https://ojs.aaai.org/index.php/AAAI/article/view/38903.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. CogACT: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024a. URL https://arxiv.org/abs/2411.19650.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators. In *International Conference on Learning Representations*, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/71639c317fb0bf398835627b4418693e-Abstract-Conference.html.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3705–3728, 2025. URL https://proceedings.mlr.press/v270/li25c.html.

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*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477. URL https://robotics-transformer-x.github.io/.

Zhenghao Peng, Wenhao Ding, Yurong You, Yuxiao Chen, Wenjie Luo, Thomas Tian, Yulong Cao, Apoorva Sharma, Danfei Xu, Boris Ivanovic, Boyi Li, Yan Wang, and Marco Pavone. Counterfactual VLA: Self-reflective vision-language-action model with adaptive reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4022–4031, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Peng_Counterfactual_VLA_Self-Reflective_Vision-Language-Action_Model_with_Adaptive_Reasoning_CVPR_2026_paper.html.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.012. URL https://www.roboticsproceedings.org/rss21/p012.html.

Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/294e09f267683c7ddc6cc5134a7e68a8-Abstract.html.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Jiayuan Gu, Zhigang Wang, Yan Ding, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action models. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.011. URL https://www.roboticsproceedings.org/rss21/p011.html.

Hao Shi, Weiye Li, Bin Xie, Yulin Wang, Renping Zhou, Tiancai Wang, Xiangyu Zhang, Ping Luo, and Gao Huang. MemoryVLA++: Temporal modeling via memory and imagination in vision-language-action models. *arXiv preprint arXiv:2606.09827*, 2026a. URL https://arxiv.org/abs/2606.09827.

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. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=54U3XHf7qq.

Ajay Sridhar, Jennifer Pan, Satvik Sharma, and Chelsea Finn. MemER: Scaling up memory for robot control via experience retrieval. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.20328.

Marcel Torne Villasevil, Andy Tang, Yuejiang Liu, and Chelsea Finn. Learning long-context diffusion policies via past-token prediction. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 1744–1755. PMLR, 2025. URL https://proceedings.mlr.press/v305/villasevil25a.html.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2564–2575, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1b113258af3968aaf3969ca67e744ff8-Abstract.html.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2c37c5bcef24b9541550261dcd63261b-Abstract-Conference.html.

Cheng Yin, Wang Xu, Junpeng Yang, Sikyuen Tam, Hanyu Liu, Yuan Yao, Xiangrui Zeng, Junbo Cui, Yequan Wang, Zhouping Yin, and Yankai Lin. SimpleMemVLA: A Simple but Effective Native-Video Memory for Vision-Language-Action Models, 2026. URL https://arxiv.org/abs/2609.05533.

Yanyan Zhang, Disheng Liu, Kai Ye, Chaoda Song, Xinpeng Li, Mohsen Hariri, Vikash Singh, Yu Yin, and Vipin Chaudhary. Imagining recovery: Inference-time counterfactual realignment for vision-language-action models. *arXiv preprint arXiv:2608.14822*, 2026. URL https://arxiv.org/abs/2608.14822.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé, III, Andrey Kolobov, Furong Huang, and Jianwei Yang. TraceVLA: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/8667f264f88c7938a73a53ab01eb1327-Abstract-Conference.html.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Model Variants and Implementation Details

A.1 Overview of evaluated variants

MemAlign specifies a history-to-action adaptation framework whose supervised interface depends on the underlying policy. The experiments use separate adaptations for RememberColor (RC), ShellGameTouch (Shell), and RoboMME. Table 5 summarizes these implementations. In particular, the RC experiments use one fixed RC5 deployment bundle across RC3, RC5, and RC9, whereas Shell uses a separately trained policy. The matched Shell ablation is also distinct from the selected Shell system used in the main controlled evaluation.

Table 5: Implementation scope of the evaluated adaptations. "Native" denotes the action-generation interface of the corresponding base policy. CF denotes counterfactual supervision.

Variant	Adapted action pathway	Evaluation scope
RC5	History-aware modulation; RGB localization; metric clean-action fuser	Fixed weights, calibration, and semantic context on RC3/RC5/RC9
Shell	History-aware modulation in the native diffusion expert; existing static LoRA retained	Main Shell O/R/E evaluation
Shell A–D	Static LoRA or history-aware modulation; action fitting with or without CF supervision; visual rebinding in B–D	Four training-matched, 2,000-update ablation arms
RoboMME V4	History-aware modulation and a residual at the native noise-prediction interface	One shared adaptation across all 16 RoboMME tasks
MemER adaptation	History-matched subgoal supervision and native flow-policy action supervision	One shared high-/low-level checkpoint combination across all 16 tasks

A.2 History-aware low-rank modulation

Let m_i^p and m_i^c denote perceptual and cognitive memory tokens recorded at time $t_i < t$. The controller projects each modality to a common space and adds modality and relative-age embeddings:

$$\widetilde{m}_i^v = P_v m_i^v + e_v + \mathrm{PE}(t - t_i), \qquad v \in \{p, c\}. \tag{8}$$

Here, PE is a sinusoidal temporal encoding. A Transformer encodes the valid tokens and learned pooling produces a history control vector z_t^{hist}. The controller has hidden dimension 256, two norm-first Transformer layers, eight attention heads, feed-forward dimension 512, GELU activations, and zero dropout. Empty history is represented by a learned null context, with a separate validity mask identifying the absence of evidence.

For an adapted linear map with input $u \in \mathbb{R}^{d_{\text{in}}}$,

$$y = W_{\text{ref}}u + b + s_h B_h[\gamma(H_{<t}) \odot (A_h u)], \qquad \Delta W_{\text{hist}} = s_h B_h \,\text{diag}(\gamma) A_h, \tag{9}$$

where $A_h \in \mathbb{R}^{r \times d_{\text{in}}}$ and $B_h \in \mathbb{R}^{d_{\text{out}} \times r}$. When an existing static adapter is present, $W_{\text{ref}} = W_0 + s_s B_s A_s$; otherwise, $W_{\text{ref}} = W_0$. The MemoryVLA adaptations use rank 8 and alpha 8 at 34 mappings: the two perceptual/cognitive projections and four linear mappings in each of action-DiT blocks 16–23.

The coefficient path compares the history control vector with its null reference before bounded modulation. The implementations include a direct history-minus-null coefficient path and an anchored coefficient bank with a frozen orthogonal sketch and a trainable residual. The selected Shell v9 coefficient residual is linear. An explicit history-validity mask enforces

$$\gamma(\varnothing) = 0, \qquad \Delta W_{\text{hist}}(\varnothing) = 0. \tag{10}$$

This property removes the additional conditional increment. Existing static adaptation and the policy's native memory pathway need not disappear. Moreover, bounded coefficients alone do not bound the norms of the trainable low-rank matrices.

The history control vector is evaluated once for an action decision and reused throughout denoising, including both classifier-free-guidance branches. For the controlled Shell v9 evaluation, the five cue frames are encoded at the step-8 handover and supply an explicit cue context reused after handover; the native memory pathway continues to process the rollout.

A.3 Visual rebinding in RememberColor

The RC implementation uses temporal visual correspondence to connect a history-specified target to the current scene. The notation $\hat{g}_t$ in the main text denotes the target selected by historical evidence; it does not require a separately supervised discrete identity classifier. The localization interface receives current RGB, a buffer of historical RGB, historical timestamps, and validity masks. It returns an image-space target estimate $\hat{u}_t$, confidence κ_t, localization uncertainty, and a correspondence-validity flag.

The temporal localizer combines a color-preserving reference path with a learned feature-matching path. The reference path uses chromaticity, saturation, and temporal changes to identify the cue-relevant appearance. The learned path compares a history-derived appearance prototype with current visual features, producing a spatial score map. A compact convolutional stem uses a 5×5 stride-2 convolution, a 3×3 stride-2 convolution, normalization and GELU, followed by depthwise and pointwise convolutions. Spatial probabilities provide a coordinate estimate and an entropy-based confidence. This module consumes visual evidence rather than simulator target coordinates at deployment.

For calibrated intrinsics K and a camera-to-base transform (R, b), the ray through homogeneous pixel $\widetilde{u}_t$ is

$$p(\lambda) = b + \lambda d_t, \qquad d_t = RK^{-1}\widetilde{u}_t. \tag{11}$$

Intersecting the ray with the calibrated table plane $n^\top p + d_0 = 0$ gives

$$\lambda^* = -\frac{n^\top b + d_0}{n^\top d_t}, \qquad \hat{p}_t^{\text{target}} = b + \lambda^* d_t. \tag{12}$$

Projection validity includes finite geometry, a usable ray–plane intersection, and valid visual correspondence. These checks prevent an invalid location from entering the executed action.

The metric feature vector contains target position (3 dimensions), target-relative TCP displacement (3), proprioception (7, with translation in the robot base frame), confidence (1), and the diagonal of the metric localization covariance (3), for 17 dimensions in total. Concatenating these features with the flattened 4×7 pre-fusion action chunk produces a 45-dimensional input. The 17 metric features

are standardized using statistics fitted to the training split and then fixed; dimensions with standard deviation at most 10^{-6} use unit scale.

The fuser has two width-256 SiLU hidden layers and a zero-initialized output layer. Its clean-action residual is

$$\Delta_\psi(A_t^{\mathrm{pre}}, z_t^{\mathrm{metric}}) = r_{\max} \tanh\left(\frac{F_\psi(A_t^{\mathrm{pre}}, z_t^{\mathrm{metric}})}{r_{\max}}\right), \tag{13}$$

with elementwise bound $r_{\max}$. Valid correspondence yields $\hat{A}_t = A_t^{\mathrm{pre}} + \Delta_\psi$; invalid correspondence returns A_t^{pre} explicitly. The fallback is the adapted pre-fusion action, which can still contain static LoRA or history-aware modulation contributions.

Teacher target positions are used to construct training action labels. They are not substituted for RGB-derived target locations at inference. During fuser training, its action input is generated by the selected adapted policy, while the expert action is the regression label. The RC5 deployment bundle, including the localizer, calibration, and five-class semantic context, is kept fixed for RC3 and RC9; no target-task class mask is added. Shell v9 does not use this metric fuser or an additional action-output head.

A.4 RoboMME noise-prediction interface

The RoboMME V4 variant uses the 34 rank-8 conditional mappings, with the static branch disabled, and adds a residual to native noise prediction:

$$\epsilon_{\mathrm{V4}} = \epsilon_{\mathrm{native}} + \rho R_\omega(A_t^{(k)}, k, \mathrm{slot}, o_t, H_{<t}). \tag{14}$$

The deployment strength $\rho = 0.50$ also scales the conditional adaptation. The native prediction term therefore denotes the denoiser with its selected history-aware adapter active at that strength.

At most 16 strictly earlier history items are retained: four initial anchors, eight recent items, and four allocated from the intervening history. Cognitive features are projected from 4,096 to 256 dimensions. Perceptual and history-control features are also projected to 256 dimensions, then summed and layer-normalized. The resulting context, a 64-dimensional diffusion-time embedding, a 64-dimensional action-slot embedding, and a seven-dimensional noised action feed a network with two width-256 SiLU hidden layers. Its output is zero-initialized, bounded by $2\tanh(\cdot/2)$, and masked by history validity. This residual network introduces no additional proprioception input. The adaptation contains 4,639,495 trainable parameters and operates inside the native denoiser forward interface, so the classifier-free-guidance path also uses the residual.

A.5 Inference procedure and available information

At each decision, the policy receives the current observation, task instruction, robot state when available, and valid causal history. It forms the history-conditioned coefficients, samples a native action chunk, and, for the RC variant, applies valid RGB-to-metric fusion. The designated prefix of this chunk is executed before replanning. Counterfactual groups, donor labels, and teacher actions are required only during supervised training or controlled evaluation construction. Ordinary deployment supplies one available history and does not enumerate alternative histories.

The RC implementation retains a ten-frame cue buffer for visual rebinding; Shell uses five cue frames for the explicit historical context. These RGB buffers are distinct from the compressed memory tokens. Encoding the cue therefore does not imply that all raw historical images are discarded.

B Counterfactual Data and Training

B.1 Legal history groups and split construction

A controlled group at handover time t is $\mathcal{G}_t = \{(H_{<t}^{(i)}, g_i, A_i^\star)\}_{i=1}^K$. Its branches share the physical layout, current RGB, instruction, and robot state. Each history is a legal cue for its own target g_i and receives the corresponding expert action $A_i^\star$. A donor history changes the desired target; it is

not a corrupted input supervised with the original action. Historical timestamps satisfy $t_i < t$, and complete scene groups remain in the same train or development partition.

Current-state equality applies at the controlled handover. Once actions differ, subsequent observations can diverge. Later teacher or recovery samples remain valid for history-matched action fitting, but are not automatically eligible for an objective requiring identical current inputs. Group-relative alignment is applied at matched decisions.

B.2 Losses and reduction conventions

The diffusion forward process for a normalized expert action chunk is

$$A_t^{(k)} = \sqrt{\bar{\alpha}_k} A_t^\star + \sqrt{1-\bar{\alpha}_k}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{15}$$

where $\bar{\alpha}_k$ is the cumulative signal-retention coefficient. The native noise-prediction loss regresses ϵ. At a clean-action interface, history-matched fitting regresses $\hat{A}_i$ against its own $A_i^\star$ using the base policy's action normalization and the valid supervision entries.

For Shell, $A \in \mathbb{R}^{4\times 7}$ and the exact weighted squared distance is

$$d_{\text{Shell}}(A, B) = \frac{1}{40}\sum_{h=1}^{4}\left[2\sum_{j=1}^{3}(A_{hj} - B_{hj})^2 + \sum_{j=4}^{7}(A_{hj} - B_{hj})^2\right]. \tag{16}$$

The denominator is the total coordinate weight, $4(2\times 3+4) = 40$. All four actions and all seven coordinates participate in this fixed-length Shell loss; predictions are the unclipped normalized DDIM outputs. The final loss averages this per-sample distance over the sampled branches and groups.

For a group with common sampling noise ξ, define $\overline{\hat{A}} = K^{-1}\sum_i \hat{A}_i$ and $\overline{A^\star} = K^{-1}\sum_i A_i^\star$. The centered action objective is

$$\mathcal{L}_\Delta = \mathbb{E}_{\mathcal{G}_t,\xi}\left[\frac{1}{K}\sum_{i=1}^{K} d_{\text{Shell}}\Big(\hat{A}_i - \overline{\hat{A}}, A_i^\star - \overline{A^\star}\Big)\right]. \tag{17}$$

Shell adds this term with weight 2 on handover batches. Shared diffusion noise reduces stochastic differences between branches, allowing the supervised contrast to focus on the behavioral change associated with history.

Why both fitting and centered alignment are used. Let $e_i = \hat{A}_i - A_i^\star$, $\bar{e} = K^{-1}\sum_i e_i$, and let $\|\cdot\|_W^2$ denote the quadratic distance in Equation 16. Then

$$\frac{1}{K}\sum_i \|e_i - \bar{e}\|_W^2 = \frac{1}{K}\sum_i \|e_i\|_W^2 - \|\bar{e}\|_W^2 = \frac{1}{2K^2}\sum_{i,j}\|e_i - e_j\|_W^2. \tag{18}$$

Thus centered alignment emphasizes agreement with the expert's within-group action changes. By itself, it is insensitive to a common action offset across all branches; matched fitting constrains that offset. When action fitting already uses the same samples and distance, the extra term changes the relative weight of group contrasts rather than adding a new action label. These identities describe the quadratic Shell objective and do not assume that every native policy uses the same distance kernel.

B.3 RC5 demonstrations and two-stage continuation

The RC5 continuation dataset contains 64 training scenes and 16 development scenes, with 560 and 141 action samples, respectively. Training reset seeds are 2026101000–2026101063 and development seeds are 2026201000–2026201015. Collection uses MIKASA 0.0.5, RGB observations, and `pd_ee_delta_pose` control. The collector stores the historical cue, current RGB, TCP proprioception, causal time, and four-action teacher chunks.

For target-relative TCP displacement Δp, the Cartesian teacher uses

$$u_{xyz} = \text{clip}\left(\frac{\Delta p + (0, 0, 0.025)}{4\times 0.1}, -0.5, 0.5\right). \tag{19}$$

Table 6: Active objectives of the MemoryVLA adaptation stages. The general weighted objective does not imply that every term is active in every stage.

Stage	Active objective	Updated components
RC5 adapter continuation	Noise-prediction MSE	Conditional A_h, B_h
RC5 fuser continuation	Normalized clean-action MSE	Existing action fuser
Shell continuation	DDIM action fitting; centered alignment with weight 2 on handover batches	Conditional A_h, B_h, coefficient bank, controller
RoboMME V4	Masked noise MSE; clean-action L_1 with weight 0.1 for $k < 50$	History-aware adapter and denoising residual

A 0.01 m deadband is applied before conversion. The command is repeated over four actions, rotation is zero, and the gripper remains open. Collection observes ten cue steps. Every fourth training scene includes small prescribed motions at cue steps 1–3 to cover changes in TCP position; development cue actions are zero. These collection perturbations are separate from the stationary controlled-test protocol. The collector retains post-success samples to cover slowing and holding behavior. Frozen visual and memory features are cached alongside RGB-derived metric features.

The final conditional stage starts from an existing RC5 adapter, retains its memory modules and conditional input matrices A_h, zero-initializes B_h, and updates only the conditional low-rank matrices. The base policy and memory encoder remain frozen. After selection, an existing action fuser is continued with fixed upstream policy predictions. Table 7 lists the two stages; their 2,400 updates exclude the earlier adapter and fuser training.

Table 7: Final RC5 continuation stages. Batch sizes count action samples.

Setting	History-aware modulation	Fuser continuation
Updates / selected update	1,200 / 1,200	1,200 / 1,200
Batch size	8	64
Learning rates	$A_h : 10^{-5}$; $B_h : 2.5 \times 10^{-4}$	5×10^{-4}
Optimizer	AdamW, $(0.9, 0.999)$	AdamW, $(0.9, 0.999)$
Weight decay / clip norm	10^{-4} / 1	10^{-4} / 1
Schedule	Constant	Constant
Training target	Noise at $k \in \{0, 10, \ldots, 90\}$	Normalized clean action
Validation	Every 200 updates	Every 200 updates
Selection	Minimum development DDIM action MSE	Minimum development DDIM action MSE

Torch and sampling seeds are 2026090801 and 2026090802. Each validation call uses generator seed 202609081234 and 32 fixed samples: the first and middle action samples from each development trajectory. Sampling uses FP32, ten DDIM steps, and classifier-free-guidance scale 1.5. Fuser inputs are generated once with the selected conditional policy; independent Gaussian noise with standard deviation 0.05 is added to these normalized input actions during training. Expert labels are not used as upstream predictions.

B.4 Shell cues, recovery data, and continuation

The Shell teacher dataset contains 64 training layouts with seeds 2026090900–2026090963 and 12 development layouts with seeds 2026091900–2026091911. Each layout includes all three target identities, yielding 192 training and 36 development trajectories. A render-only cue displays the ball associated with the selected target during steps 0–4; the physical ball pose is restored before the environment advances. The first eight actions are zero. RGB and proprioception are checked for equality across target branches at handover.

The teacher follows a clearance-then-translation strategy. It uses a height offset of 0.14 m while lateral distance exceeds 0.06 m and 0.06 m otherwise. During lateral motion, XY commands are held at zero until the required clearance is reached. Four-action chunks are normalized, clipped to the representable action range, and decoded with the deployment statistics before execution, with a 90-step horizon.

The selected v9 continuation also uses 24 recovery trajectories from the three target branches of eight existing training layouts: the odd seeds from 2026090901 through 2026090915. The v6 update-600 policy drives these rollouts until the step-40 teacher takeover. Teacher labels therefore cover states visited by the learned policy. Development layouts never enter this pool.

Each update samples two complete three-target groups. Recovery data are chosen with probability 0.7; otherwise the original teacher pool is used. Handover samples have probability 0.2, while other samples are chosen by a uniform trajectory fraction. All branches within a group share the initial diffusion noise. Handover groups receive both matched action fitting and the centered loss; later samples receive their own action labels without assuming their current states remain equal.

Table 8: Shell v9 continuation configuration. Update counts start after the v6 update-600 initialization.

Setting	Value
Selected continuation update	200; compared with update 400 on development rollouts
Batch	Two three-target groups; six action samples
Trainable modules	Conditional A_h, B_h, coefficient bank, history controller
Frozen modules	Pretrained backbone, memory reader, existing static LoRA
Learning rates	A_h : 10^{-5}; B_h/coefficient bank: 10^{-4}; controller: 2×10^{-5}
Optimizer / schedule	AdamW / constant rates
Weight decay / clip norm	10^{-4} / 1
CPU and CUDA seed	2026090901
Memory dropout	0
Action generation	FP32; DDIM-10; CFG 1.5; 4×7 actions
Diagnostics / selection	Every 100 updates / closed-loop results on 12 development layouts

B.5 RoboMME collection, action alignment, and optimization

The benchmark planners collect 64 successful trajectories per task for all 16 tasks, with 32 easy, 16 medium, and 16 hard trajectories per task. The resulting corpus contains 1,024 trajectories and 493,121 frames. For task index q, episode index e, and retry index r, candidate seeds follow $260906000 + 10^6 q + 100e + r$ and are checked against official train, validation, and test metadata. Collection permits up to eight fresh-seed attempts, each with a 1,200-second timeout. Successful-trajectory filtering applies to training collection; failed evaluation episodes remain in the denominator.

Each trajectory stores front RGB, end-effector state, absolute commands, gripper state, timestamps, and demonstration-frame flags. Nonfinite actions, trajectories shorter than 17 frames, and trajectories containing only demonstration frames are rejected. The stored observation at index t follows command t, so future action supervision begins with command $t + 1$. Translation is converted relative to the corresponding end-effector state, Euler differences are wrapped, and the released gripper representation is retained. Terminal and demonstration-only commands are masked.

Trajectory indices satisfying $e \bmod 20 \in \{3, 6, 9, 12\}$ form the held-out partition. Training uses 816 trajectories. The fixed offline development reference contains 512 deterministic samples from 105 held-out trajectories; the other 103 held-out trajectories are unused for selection. Sampling first chooses a task uniformly, then one of its training trajectories, and then a valid execution time. Each sample has at most 16 strictly earlier history positions and a 16-action prediction target.

At optimizer update s, the implemented schedule is

$$\eta_s = \eta_0 \min(1, s/50) \left[0.2 + 0.4 \left(1 + \cos\Big(\frac{\pi s}{15000}\Big)\right)\right]. \tag{20}$$

After offline checkpoint selection, adaptation strength is selected with four development episodes per task. This inference scaling introduces no additional optimizer updates. Offline selection and online strength selection are separate from the final 800-episode test evaluation.

B.6 Ordinary action-supervision control and matched ablations

The ordinary action-supervision control initializes a new memory reader, injector, history controller, coefficient residual, and rank-8 adapter on the frozen released MemoryVLA backbone. It

Table 9: RoboMME V4 optimization and model selection.

Setting	Value
Training / selected updates	15,000 / 14,500
Global batch / world size	16 action samples / 1
Training seed / precision	260906 / BF16 autocast
Optimizer	AdamW, $\beta = (0.9, 0.95)$
Learning-rate parameter	$\eta_0 = 10^{-4}$
Weight decay / clip norm	0.01 / 1
Diffusion indices	Uniform integer in $[0, 99]$
Objective	Masked noise MSE plus 0.1 clean-action L_1 when $k < 50$
Offline selection	Lowest loss on 512 fixed development samples; every 500 updates
Deployment strength	0.50; chosen on a separate 64-episode development set

has 34,778,569 trainable parameters, without a previously trained static LoRA or the RC metric fuser. It is trained on legal same-layout history groups, with each branch paired with its own action label. Explicit counterfactual alignment weights are zero. Consequently, this control tests what ordinary action regression can learn from history-matched data; it does not remove alternative histories from training.

Table 10: Data for the ordinary action-supervision control. Development and test columns count scenes per history condition. These test sets are separate from the main 50-scene evaluations.

Task	Train layouts	Targets/layout	Train branches	Dev	Test
RC3	32	3	96	6	30
RC9	24	9	216	9	27
Shell	64	3	192	6	30

Each update uses two complete groups, with handover probability 0.5 and shared sampling noise within a group. The training budget is 2,000 updates with initialization seed 2026091211, AdamW betas $(0.9, 0.999)$, weight decay 10^{-4}, gradient clipping at norm 1, and constant learning rates: 2×10^{-5} for the reader/injector/controller, 10^{-4} for the coefficient residual and B_h, and 10^{-5} for A_h. Development selection compares updates 500, 1,000, 1,500, and 2,000 by the mean of Original success and donor-first completion, breaking ties by Original success and then the earlier update. RC3, RC9, and Shell select updates 1,500, 1,000, and 1,500. This selection diagnostic does not replace exclusive completion in the controlled evaluation.

The four-arm Shell ablation instead holds the backbone, training layouts, data/noise schedule, update budget, and development selection rule fixed while varying static LoRA versus history-aware modulation and the inclusion of CF supervision. Visual rebinding is enabled in B–D and disabled in A. Every arm receives 2,000 updates, and evaluation uses 50 paired scenes with one training initialization. This matched comparison, reported in Section E.3, is distinct from the ordinary supervision system above and from the warm-started Shell v9 continuation.

C Evaluation Protocols and Metrics

C.1 Controlled history interventions

For an original target g_i and a different donor target g_j in the same layout, the three conditions are

$$\mathrm{O} : (x_t, H^{(i)}_{<t}) \longrightarrow g_i, \tag{21}$$

$$\mathrm{R} : (x_t, H^{(j)}_{<t}) \longrightarrow g_j, \tag{22}$$

$$\mathrm{E} : (x_t, \varnothing) \longrightarrow \text{diagnostic behavior}. \tag{23}$$

Both the released MemoryVLA baseline and MemAlign remain stationary during the cue-observation prefix: ten steps for RC and eight for Shell. Policy actions begin afterward. Current RGB, instruction, and robot state are matched at handover, and paired conditions share the initial action-generation noise. Each condition subsequently follows its own closed-loop observations.

This stationary-prefix protocol separates observing the cue from acting on it. It differs from source evaluations that permit policy execution during the cue period. Published scores under those protocols provide benchmark context, while the local baseline supplies the comparison for our controlled interventions. The two types of scores are not pooled into a common estimate.

Erasure semantics. Erasure removes the target cue without supplying an alternative target. The original target remains a reporting reference, not an action label designated as uniquely correct under absent evidence. In Shell v9, history is cleared before every subsequent prediction. In the ordinary supervision Shell control, it is cleared at handover and later observations can accumulate. In RC, removing the historical target also disables its target-dependent metric fusion. These implementations determine what information remains available and must be considered when interpreting E. There is no preferred ordering of erased-history success rates.

C.2 Completion and scene-paired metrics

Let $S_{n,c}$ be the set of objects completed during the evaluated execution window of scene n under condition c. Let $g_{n,c}$ be the original target for O/E and the donor target for R. Exclusive and cumulative target completion are

$$Y^{\mathrm{excl}}_{n,c} = \mathbf{1}\{S_{n,c} = \{g_{n,c}\}\}, \qquad Y^{\mathrm{cum}}_{n,c} = \mathbf{1}\{g_{n,c} \in S_{n,c}\}. \tag{24}$$

The primary MIKASA metric is exclusive completion, averaged across scenes. Cumulative completion can also count episodes that complete additional objects and therefore does not by itself establish target-specific selection. Neither metric is replaced by proximity in a trajectory projection or by a first-contact marker.

Joint O/R completion measures success in both interventions for the same scene:

$$\mathrm{Joint}_{\mathrm{O/R}} = \frac{1}{N}\sum_{n=1}^{N} Y^{\mathrm{excl}}_{n,\mathrm{O}} Y^{\mathrm{excl}}_{n,\mathrm{R}}. \tag{25}$$

It is computed from scene-matched outcomes and cannot be recovered by multiplying the two marginal success rates.

We additionally partition each episode into five mutually exclusive outcomes: original only, donor only, other only, multiple, or none. Under R, singleton completion of the original target and donor target are distinguished; other only denotes a different singleton. Under O and E, any non-original singleton is counted as other only. Multiple denotes $|S_{n,c}| > 1$, and none denotes $S_{n,c} = \varnothing$. The categories therefore sum to the full episode count and do not discard unsuccessful rollouts.

C.3 Execution settings and sample counts

Task coverage. The controlled MIKASA-Robo evaluation uses RememberColor3, RememberColor5, RememberColor9, and ShellGameTouch. The RememberColor tasks require selecting one of three, five, or nine objects according to an earlier color cue. ShellGameTouch requires remembering which cup was associated with an earlier ball cue. The stationary cue-observation protocol is specified in Appendix C.1.

Table 11: Local evaluation settings. O/R/E interventions apply to the controlled MIKASA experiments. RoboMME reports native benchmark task success.

Setting	RememberColor	Shell v9	RoboMME V4
Episodes	50 per task and condition	50 per condition	50 per task; 800 total
Cue prefix	10 stationary steps	8 stationary steps	Native task protocol
Predicted / executed actions	4 / 4	4 / 4	16 / 8
Environment step cap	60	90	1,300
Sampler	DDIM-10	DDIM-10	DDIM-10
CFG / precision	1.5 / FP32	1.5 / FP32	1.5 / FP32
Primary metric	Exclusive completion	Exclusive completion	Task success

Object completion requires a three-dimensional TCP-to-object distance of at most 0.05 m in RC or 0.08 m in Shell, together with the native staticness conditions. Shell v9 records completions from step 8 onward. Native stopping and truncation rules remain active; the completion set is accumulated over the actual execution window.

The local MemoryVLA-based RoboMME comparison uses all 16 tasks with inference seed 7, reset after model construction. Infrastructure errors permit at most three attempts; exhausted retries remain failures. Ordinary task failures are not retried. These rules preserve the 800-episode denominator. MemER uses the separate evaluation setup described in Section D.4.

Conventional manipulation benchmarks. LIBERO (Liu et al., 2023) evaluation covers LIBERO-10, LIBERO-Object, LIBERO-Goal, and LIBERO-Spatial, with 100 episodes per suite and 400 in total. The WidowX/Bridge setting of SIMPLER (Li et al., 2025) covers four tasks: putting a carrot on a plate, stacking a cube, putting an eggplant in a basket, and putting a spoon on a towel. Each task uses 24 episodes, giving 96 in total. Both benchmarks measure conventional task success without the controlled history interventions.

C.4 Aggregation and interpretation

Rates are computed from episode counts before rounding. Since each task has the same number of episodes within a benchmark, its macro-average equals the corresponding pooled success rate. Percentage-point changes use the unrounded count-based rates where counts are available. For manuscript aggregates supplied at two-decimal precision, differences are reported at that precision.

The reported episode rates describe the evaluated policy and scenes. They are not means over independent training runs. Shared seeds reduce differences between paired evaluation conditions but do not measure variation across training initializations. The matched Shell ablation uses one initialization; we consequently interpret its differences descriptively rather than as a multi-seed significance result.

D Adapting Counterfactual Supervision to MemER

D.1 Native hierarchical interfaces

MemER separates memory-dependent subgoal generation from low-level action execution. Denote its high-level subgoal distribution by $q_\phi(z \mid x_t, H_{<t})$ and its low-level policy by $\pi_\psi(A \mid x_t, z)$. Historical visual evidence is interpreted at the high level and communicated through the native grounded-subgoal interface. The low-level policy retains its native image, state, action, normalization, and padding conventions.

We adapt the supervision principle to these interfaces: a legal history is paired with the subgoal and action appropriate to its target, and a replaced history receives the corresponding replacement supervision. This preserves MemER's architecture. It does not require transplanting MemoryVLA's perceptual/cognitive token dimensions, conditional-LoRA wrappers, diffusion noise targets, or RC metric fuser into MemER.

D.2 History-matched subgoal supervision

For a legal history branch i, let $z_i^\star$ denote its target-consistent subgoal text. Matched high-level fitting has the form

$$\mathcal{L}_{\text{high,match}} = -\mathbb{E}_{(x_t, H_{<t}^{(i)}, z_i^\star)} \log q_\phi(z_i^\star \mid x_t, H_{<t}^{(i)}). \tag{26}$$

The subgoal text is a supervision target; it is not appended to the policy input as an answer. Counterfactual comparisons evaluate whether the target associated with the supplied history is preferred over an alternative target under the same current context. Natural memory-dependent examples and retention objectives accompany these comparisons to constrain changes to the existing subgoal and memory-writing behavior.

The high-level supervision acts on the native textual outputs. Action-level supervision acts on the low-level action policy. The adaptation does not assume that action gradients pass through discrete subgoal generation.

D.3 Native flow matching and action correspondence

MemER's low-level action policy uses a flow parameterization. With normalized expert action chunk $A^\star$, Gaussian noise ϵ, and flow time $\tau \in [0, 1]$, the native training path is

$$X_\tau = (1 - \tau)A^\star + \tau\epsilon, \qquad V^\star = \epsilon - A^\star. \tag{27}$$

Its fitting objective regresses the velocity $V_\psi(X_\tau, \tau, x_t, z)$ against $V^\star$ over valid native action entries. A clean-action estimate at this supervised interface can be written as

$$\widehat{A} = X_\tau - \tau V_\psi(X_\tau, \tau, x_t, z). \tag{28}$$

This relation provides an action-space interface for matched regression and counterfactual action comparisons; it does not replace the native inference integrator.

Each branch is supervised with its own expert continuation. Controlled comparisons share current inputs, noise, and flow time, while expert action differences specify the desired change between historical targets. Masks exclude padding and unavailable future actions. Retention terms constrain deviation from native behavior on their designated reference inputs, rather than treating erased history as a deliberately wrong action target. MemoryVLA's epsilon-prediction loss is therefore not used as a substitute for MemER's flow-velocity target.

D.4 Evaluation and scope of the transfer result

The adapted MemER policy is evaluated on all 16 RoboMME tasks with 50 episodes per task, for a total of 800 episodes. It completes 352 episodes, yielding an aggregate success rate of 44.00%. Since each task contributes the same number of episodes, this aggregate also equals the unweighted mean of the 16 task success rates.

The reported MemER reference score is 42.38%, giving a difference of 1.62 percentage points between the two reported rates. This reference follows its source evaluation protocol; the comparison does not constitute a matched local evaluation of the released and adapted policies. The result supports the applicability of MemAlign's history-to-decision supervision to a hierarchical policy with a different action interface. Coverage of all 16 tasks at evaluation does not imply that every task supplies identical types or amounts of counterfactual training data.

E Additional Results and Behavioral Analysis

E.1 Complete controlled outcome counts

Table 12 gives the full mutually exclusive outcome counts underlying the controlled MIKASA results. Every row contains 50 episodes. The designated success column is original only for O and donor only for R. E uses original only solely as a diagnostic reference.

Under O, the four-task average exclusive completion increases from 13.00% to 87.00%. Under R, donor-only completion increases from 11.00% to 87.50%. The replaced-history result is evidence of successful redirection because each successful episode completes the donor alone. Erasing history yields original-only rates of 32%, 2%, 10%, and 20% for MemAlign on RC3, RC5, RC9, and Shell, respectively. These rates characterize behavior without the cue.

The remaining failures differ by task. In RC9, MemAlign's O episodes include 13 other-only, three multiple-completion, and four no-completion outcomes. Its R episodes include 12 other-only, five multiple-completion, and three no-completion outcomes. In Shell, all five unsuccessful O episodes and all five unsuccessful R episodes have no completed object. These categories describe observable behavior; they do not identify whether an internal error originates in cue interpretation, visual correspondence, or action execution.

Table 12: Controlled MIKASA outcome counts. Columns Orig., Donor, and Other refer to singleton completion; Multi. refers to multiple completed objects. The donor-only category is distinguished only under R.

Task	Policy	History	Orig.	Donor	Other	Multi.	None
RC3	MemoryVLA	O	13	0	26	7	4
		R	15	13	14	4	4
		E	13	0	28	4	5
	MEMALIGN	O	50	0	0	0	0
		R	0	50	0	0	0
		E	16	0	27	4	3
RC5	MemoryVLA	O	1	0	9	0	40
		R	1	3	6	0	40
		E	1	0	14	0	35
	MEMALIGN	O	49	0	0	0	1
		R	0	50	0	0	0
		E	1	0	7	0	42
RC9	MemoryVLA	O	5	0	27	2	16
		R	4	1	26	1	18
		E	3	0	33	2	12
	MEMALIGN	O	30	0	13	3	4
		R	0	30	12	5	3
		E	5	0	29	1	15
Shell	MemoryVLA	O	7	0	11	0	32
		R	4	5	7	0	34
		E	9	0	13	0	28
	MEMALIGN	O	45	0	0	0	5
		R	0	45	0	0	5
		E	10	0	23	1	16

E.2 FULL LOCAL ROBOMME BREAKDOWN

Table 13 reports all 16 tasks in the local MemoryVLA comparison. The total increases from 66/800 to 87/800, or from 8.25% to 10.875% (10.88% after rounding). This is 21 additional successful episodes and an improvement of 2.625 percentage points (2.63 after rounding). The largest gains in successful-episode counts are on VideoPlaceButton ($+8$), VideoPlaceOrder ($+7$), and VideoUnmask, ButtonUnmask, and MoveCube ($+4$ each). Improvements are not uniform: several tasks decrease, and VideoRepick and InsertPeg remain at zero in both systems.

Published RoboMME scores in Table 2 follow Tables 3 and 15 of Dai et al. (2026). They retain their source protocols and are separate from this local seed-7 block. The reported MME-VLA entry is the Frame sampling + Modulator variant. The local improvement is computed against the locally evaluated MemoryVLA baseline, rather than against its published aggregate. The separate MemER result is summarized in Table 14; it uses the hierarchical supervision adaptation described in Section D.

E.3 MATCHED SHELL ABLATIONS

With visual rebinding and CF supervision fixed, replacing static LoRA with history-aware modulation (B $\rightarrow$ D) increases O/R completion from 38%/38% to 52%/56%, gains of 14/18 percentage points. With visual rebinding and history-aware modulation fixed, adding CF supervision (C $\rightarrow$ D) improves O/R completion from 44%/46% to 52%/56%, gains of 8/10 points. Joint completion also increases from 12% for C to 20% for D. Comparisons of A with B or C change visual rebinding together with supervision or parameterization, respectively, and therefore do not isolate either change.

The combined arm has the highest O, R, and Joint O/R rates in this matched comparison. The E column remains diagnostic and is not optimized for a lower value. The 90%/90% O/R result of Shell v9 belongs to a different warm-started system with recovery-data continuation and must not be substituted for arm D in this table.

Table 13: Full local RoboMME results with inference seed 7. Each cell reports successful episodes out of 50 and the corresponding percentage. Category subtotals retain all four tasks, including zero-success tasks.

Category	Task	MemoryVLA	MEMALIGN	Δ (pp)
Counting	BinFill	6 (12%)	4 (8%)	−4
	PickXtimes	11 (22%)	6 (12%)	−10
	SwingXtimes	2 (4%)	1 (2%)	−2
	StopCube	0 (0%)	2 (4%)	+4
	Subtotal / 200	19 (9.5%)	13 (6.5%)	−3
Permanence	VideoUnmask	9 (18%)	13 (26%)	+8
	ButtonUnmask	6 (12%)	10 (20%)	+8
	VideoUnmaskSwap	5 (10%)	6 (12%)	+2
	ButtonUnmaskSwap	2 (4%)	0 (0%)	−4
	Subtotal / 200	22 (11%)	29 (14.5%)	+3.5
Reference	PickHighlight	4 (8%)	6 (12%)	+4
	VideoRepick	0 (0%)	0 (0%)	0
	VideoPlaceButton	6 (12%)	14 (28%)	+16
	VideoPlaceOrder	1 (2%)	8 (16%)	+14
	Subtotal / 200	11 (5.5%)	28 (14%)	+8.5
Imitation	MoveCube	8 (16%)	12 (24%)	+8
	InsertPeg	0 (0%)	0 (0%)	0
	PatternLock	6 (12%)	4 (8%)	−4
	RouteStick	0 (0%)	1 (2%)	+2
	Subtotal / 200	14 (7%)	17 (8.5%)	+1.5
Overall / 800		66 (8.25%)	87 (10.88%)	+2.63

Table 14: Reported MemER supervision-transfer result on all 16 RoboMME tasks. MEMALIGN on MemER use 50 validation episodes per task and inference seed 7.

System	Episodes	Success (%)
MemER (published reference)	800	42.38
MemER + MEMALIGN supervision	800	44.00
Difference (percentage points)		+1.62

Table 15: Shell ablation with 2,000 updates per arm and 50 paired scenes per history condition. All values are percentages. Joint O/R is evaluated per scene using Equation 25.

Parameterization	Supervision	Visual rebinding	O	R	E	Joint O/R
A: Static LoRA	Action fitting	—	22	16	20	0
B: Static LoRA	Action fitting + CF	✓	38	38	36	4
C: History-aware modulation	Action fitting	✓	44	46	34	12
D: History-aware modulation	Action fitting + CF	✓	**52**	**56**	22	**20**

E.4 TRANSFER ACROSS CANDIDATE-SET SIZES

The RC5 bundle is evaluated directly on tasks containing three, five, and nine candidate objects. The selected history-aware adapter, localizer, metric fuser, calibration, and five-class semantic context are fixed. No RC3- or RC9-specific training or class mask is introduced. Table 16 reports exclusive completion under both legal-history conditions.

Table 16: Candidate-set transfer with one fixed RC5 deployment bundle. Each condition has 50 episodes; scores are percentages.

Task	MemoryVLA O	MemoryVLA R	MEMALIGN O	MEMALIGN R
RememberColor3	26	26	100	100
RememberColor5	2	6	98	100
RememberColor9	10	2	60	60

The gains extend to selecting the donor target, beyond completing the original task. The lower RC9 result also shows the limit of this transfer: more competing candidates remain challenging. This experiment varies candidate-set size within the evaluated task family.

E.5 CONVENTIONAL MANIPULATION BREAKDOWN

Table 17: Conventional manipulation results, without controlled history interventions. Values are successful episodes / evaluated episodes.

Suite or task	MemoryVLA	MEMALIGN	Δ (pp)
LIBERO-10	93/100	94/100	+1.00
LIBERO-Object	99/100	97/100	−2.00
LIBERO-Goal	97/100	98/100	+1.00
LIBERO-Spatial	98/100	96/100	−2.00
Overall LIBERO	387/400	385/400	−0.50
Put a carrot on a plate	18/24	19/24	+4.17
Stack a cube	9/24	9/24	0.00
Put an eggplant in a basket	24/24	23/24	−4.17
Put a spoon on a towel	18/24	18/24	0.00
Overall SIMPLER (Bridge)	69/96	69/96	0.00

LIBERO aggregate success changes from 96.75% to 96.25%, while both systems achieve 71.88% on SIMPLER (Bridge) after rounding. The aggregate results are close in these samples, with gains and losses on individual tasks. This comparison describes the conventional-manipulation configurations reported in the main paper.

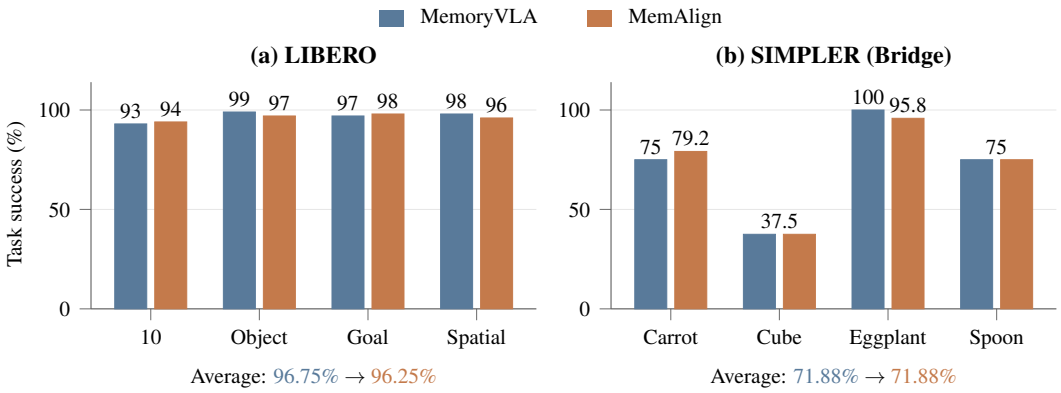


Figure 5: **Results on general manipulation benchmarks.** Task success on four LIBERO suites (100 episodes each) and four SIMPLER (Bridge) tasks (24 episodes each). Averages are computed from the unrounded episode counts; full task names and counts are given in Table 17.

E.6 TRAJECTORY EXAMPLES AND BEHAVIORAL INTERPRETATION

Figure 4 shows RC3 scene 00, whose original and donor targets are objects 1 and 2. MemoryVLA completes object 0 in all three conditions. With the fixed RC5 adaptation, O completes object 1, R completes object 2, and E completes object 0. The current inputs coincide at handover; the different trajectories subsequently induce different observations.

This example illustrates target redirection associated with changing the available historical evidence. The aggregate donor-only counts establish its frequency over the evaluated scenes; the single trajectory is illustrative. Neither the trajectory nor the outcome table alone proves that a particular internal retrieval operation succeeded or failed. Shell provides complementary behavioral evidence through its native diffusion action pathway without the RC metric fuser.

The complete trajectory collections for RC3, RC5, and RC9, covering all 50 evaluated scenes per task, are provided in the supplementary material. They contain successful, wrong-object, multiple-completion, and no-completion rollouts rather than a selection based on success. RC3 and RC9 panels display the recorded completed-object sets; their full counts agree with Table 12. The RC5 collection comes from the same 50-scene evaluation as the RC5 rows of that table. XY projections and height traces are shown together; completion remains defined by the recorded three-dimensional distance and staticness criteria.

Figure 6 adds three Shell examples from the 50-scene evaluation: scene–target pairs $(2026092900, 0)$, $(2026092901, 1)$, and $(2026092902, 2)$. These are the first three pairs in the fixed scene ordering, not examples selected for a particular outcome. Their O/R/E traces and the original-history baseline traces match the corresponding evaluation records. Other target branches from the same layouts are not included in this figure because they are outside the reported 50-pair subset.

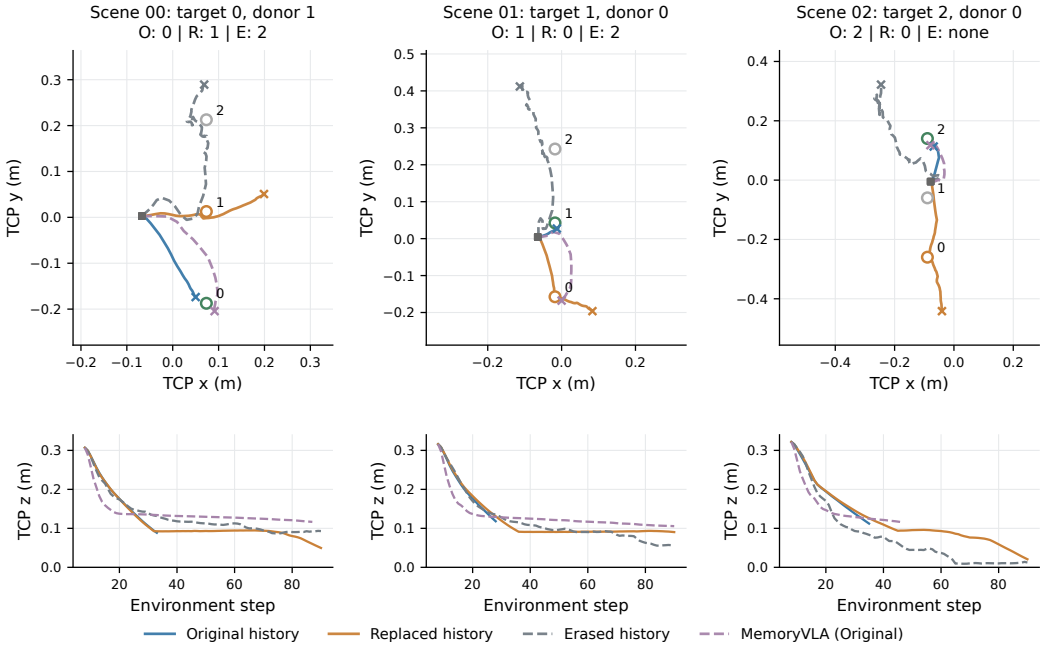


Figure 6: Shell trajectories for three scene–target pairs in the reported 50-scene evaluation. O/R/E denote the adapted policy under original, replaced, and erased history. The additional baseline curve uses original history. Panel headings list the recorded completed-object sets for O/R/E, with "none" indicating no completed cup; they are not first-contact labels. Squares mark the handover and crosses mark final TCP positions. Upper panels show XY motion and lower panels show height. Completion uses the full three-dimensional criterion and native staticness conditions.

F Reproducibility and Limitations

F.1 Experimental units, model selection, and resources

The configuration tables distinguish training updates, branch samples, scene layouts, and evaluation episodes. RC5 uses one selected deployment bundle across candidate-set sizes. Shell v9, the training-matched Shell arms, RoboMME V4, and MemER use separate adaptations, with the model-selection procedures described in their respective sections. Scene-level grouping prevents target branches from crossing the added-data train/development boundary.

Experiments used a resource pool of six NVIDIA L40S GPUs. The final RC5 continuation and RoboMME V4 training configurations use one training process; the resource-pool size does not denote the GPU count of every run. FP32 is used for the reported MemoryVLA-based action sampling, whereas RoboMME V4 training uses BF16 autocast. The RC5 and Shell continuation counts exclude the training that produced their inherited adapters. Full end-to-end training cost, raw-history memory consumption, and per-decision latency are not quantified here, so update counts are not presented as GPU-hours or runtime speedups.

F.2 Scope of the evidence

Controlled cue timing. The MIKASA experiments use a designated cue period followed by a controlled handover. This enables matched-history interventions but does not establish performance when relevant evidence arrives at arbitrary times, must be retained indefinitely, or is interleaved with unconstrained robot motion.

System-level versus component-level attribution. The RC system includes explicit RGB-to-metric grounding. Its improvement therefore reflects the deployed combination of memory-conditioned adaptation, visual correspondence, and action fusion. In the matched Shell ablation, B–D hold visual rebinding fixed: B versus D isolates history-aware modulation with CF supervision, and C versus D isolates CF supervision with history-aware modulation. Comparisons against A also change visual rebinding and do not isolate its contribution.

Geometry and retained visual history. The RC visual rebinding module depends on usable visual correspondence and a calibrated projection into the robot frame. Its explicit fallback preserves the pre-fusion action when correspondence is invalid; it does not guarantee correct target selection. Retained RGB cue buffers also have storage and processing costs beyond compressed memory tokens. Broader camera and scene changes, longer histories, and measured inference costs remain relevant directions for evaluation.

Conventional manipulation and deployment. The LIBERO and SIMPLER (Bridge) aggregates describe the reported samples and configurations, rather than proving absence of forgetting across all possible tasks or adaptations. All results in this appendix concern the stated simulation evaluations; robustness in real-robot deployment is not established by these experiments.