FAME-VLA: FULL-HISTORY ADAPTER FOR MEMORY -ENHANCED VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained Vision-Language-Action (VLA) models have shown strong promise for robot manipulation. However, most existing VLAs condition primarily on the current observation or a short window of recent observations, limiting their ability to resolve history-dependent ambiguities in long-horizon tasks. To address this, we introduce FAME-VLA, a lightweight full-history adapter that equips pretrained VLAs with dual-stream memory for history-aware action generation. Our adapter combines two complementary memory branches: full-history temporal memory, which compresses the complete observation history into a compact task context, and persistent initial-scene memory, which preserves access to spatial details that may become occluded or altered during execution. These history-aware representations are injected exclusively into the action expert, making our adapter a lightweight, plug-and-play module for pretrained VLAs. To efficiently scale to full observation histories, we precompute history features once per trajectory during training. Experiments on simulation benchmarks and real-world manipulation tasks demonstrate consistent improvements across pretrained VLA backbones, achieving average success rates of 56.11% with StarVLA and 64.89% with π_0 on RMBench. Further analyses demonstrate the complementary roles of the two memory branches and the effectiveness of full-history conditioning, while history sharing provides up to a $4.13\times$ training speedup over independently encoding each observation prefix.

1 INTRODUCTION

Learning generalist policies that can adapt to diverse manipulation tasks remains a central challenge in robotics. Pretrained vision-language-action (VLA) models (Black et al., 2024; Liu et al., 2025; Intelligence et al., 2025; NVIDIA et al., 2025) provide a promising foundation by combining visual-semantic representations with manipulation skills learned from diverse robot demonstrations. Through downstream fine-tuning, these capabilities can be adapted to new tasks, making pretrained VLAs a practical starting point for extending robot capabilities. However, many existing VLAs condition action prediction primarily on the current observation or a short window of recent frames.

This short-context conditioning can be insufficient even for history-dependent tasks. In these tasks, visually similar observations may require different actions depending on what occurred earlier. For example, returning an object to its original location requires recalling its position before it was moved. The information needed to resolve this kind of ambiguity may no longer be available in the current scene. Without access to the relevant history, policies may repeat completed steps, undo previous actions, or return objects to incorrect locations.

Existing methods attempt to incorporate history through hierarchical planning (Torne et al.; Chen et al., 2026; Sridhar et al., 2025) or multi-frame conditioning (Li et al., 2025; Jang et al., 2025). Planning provides explicit temporal structure but requires additional supervision, whereas directly expanding the visual context increases computation (Koo et al., 2026) and can introduce spurious correlations (Mark et al., 2026). To reduce this overhead, compact alternatives selectively retain keyframes (Zeng et al., 2026; Wei et al., 2026), visual traces (Zheng et al., 2025b), or retrieved memories (Shi et al., 2026). However, selection may discard relevant evidence and sometimes requires event supervision (Yang et al., 2026). Learned memory can encode longer histories, but compact temporal representations may lose spatial details needed to recover earlier scene config-

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.

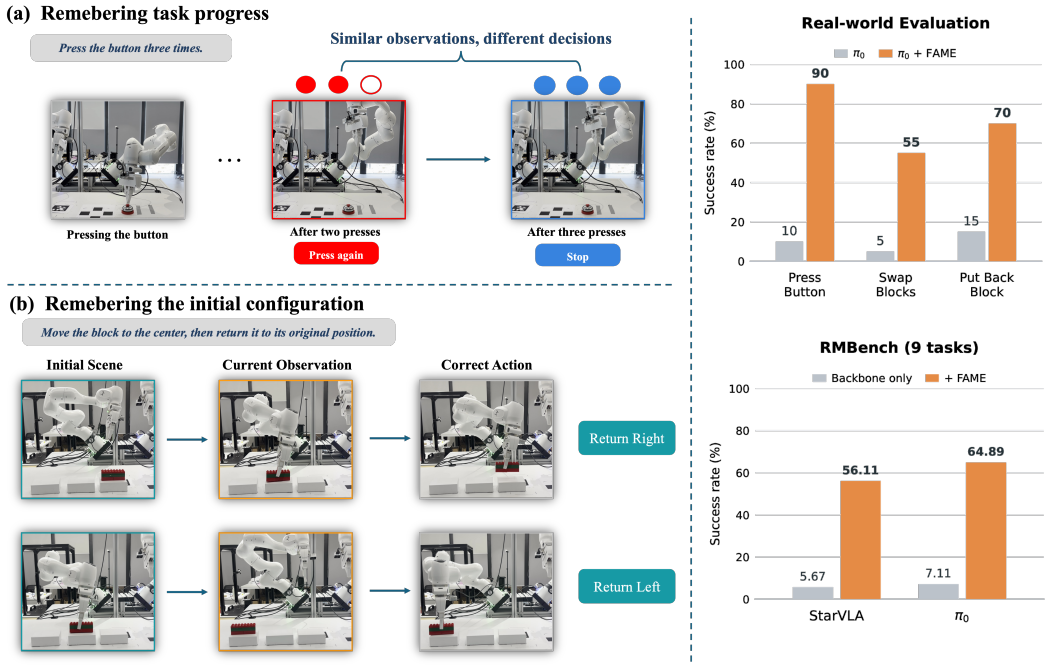


Figure 1: The proposed **FAME-VLA** integrates full-history context and initial-scene memory to resolve visual aliasing in memory-dependent manipulation, achieving superior success rates in both simulation and real-world experiments.

urations. Moreover, truncated backpropagation in recurrent models (Li et al., 2026; Cherepanov et al., 2026) and frozen encoders in two-stage training (Zhou et al., 2026; Wang et al., 2026b) limit how directly downstream action supervision shapes information retained from distant observations. These trade-offs motivate a framework that connects the full observed history to action prediction in pretrained VLAs, preserving complementary temporal and spatial information while jointly learning how to encode and use it under action supervision.

To address these needs, we introduce FAME-VLA, a lightweight modular framework that bridges full observation histories and pretrained VLAs through dual-memory integration. The framework combines a causal temporal memory that encodes the full observed history with a persistent initial-scene memory queried by the current observation. Together, the two branches provide evolving temporal context and access to original spatial details that compact temporal representations may otherwise lose. To integrate this context with pretrained capabilities, we introduce zero-initialized modulation in the action expert, preserving the pretrained computation at initialization. We jointly optimize the memory modules and policy-adaptation parameters under the original action objective in a single stage, allowing action supervision to shape both what memory retains and how it informs action prediction. To make full-history training efficient, we process training in an episode-wise manner, allowing sampled timesteps to share history computation rather than repeatedly encoding overlapping prefixes.

Our contributions are summarized as follows:

- We propose FAME-VLA, a lightweight framework that equips pretrained VLA backbones with memory. By incorporating historical context into action prediction, the framework extends manipulation capabilities to tasks that require temporal reasoning.
- We develop a dual-memory design that combines full-history temporal memory with persistent initial-scene memory. The two branches capture evolving interactions and preserve the original scene configuration, providing complementary context for resolving ambiguities about task progress and prior object locations.
- We evaluate FAME-VLA on simulated and real-world memory-dependent tasks, achieving absolute average success-rate gains of up to 57.78% on RMBench and 61.67% in real-

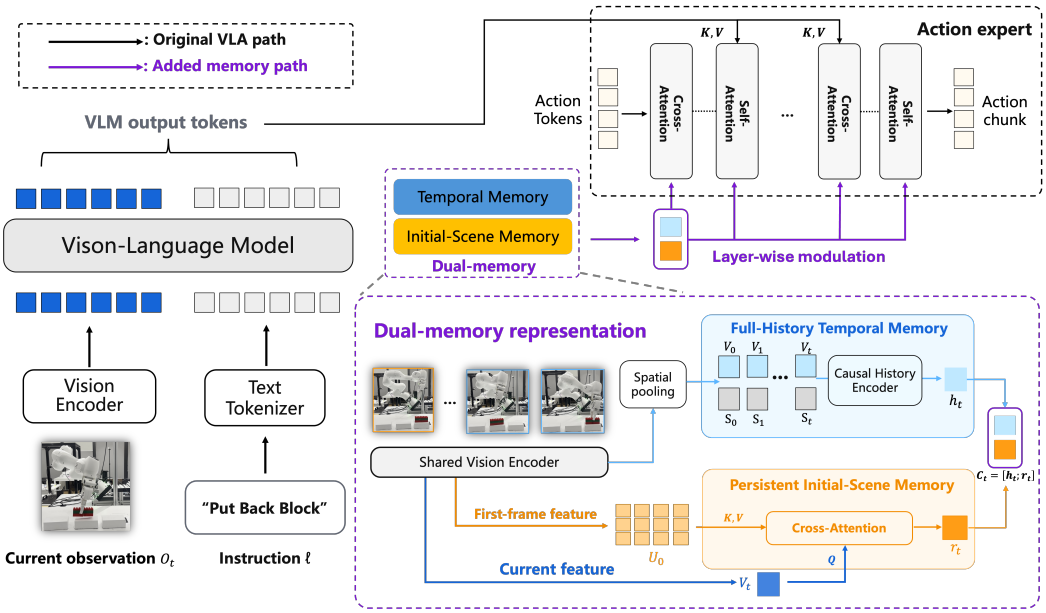


Figure 2: **Overview of FAME-VLA.** Our framework augments a pretrained VLA with two complementary memory branches. Full-history temporal memory encodes the observed history through learned spatial pooling and a causal temporal encoder, while persistent initial-scene memory retrieves spatial details from the initial observation using the current visual features as queries. The two memory representations jointly condition the action expert through zero-initialized, layer-wise adaptive normalization, without influencing VLM input tokens.

world experiments over backbone-only baselines. Further analyses examine full-history encoding, memory complementarity, and training efficiency through episode-wise history sharing.

2 Related Work

Vision-Language-Action model (VLA) Imitation learning enables robotic manipulation from expert demonstrations (Shafiullah et al., 2022; Chi et al., 2024; Lee et al., 2024; Black et al., 2025), but task-specific data limit generalization. Vision-language-action (VLA) models transfer pretrained visual-semantic knowledge to robotic control, predicting discrete action tokens autoregressively (Brohan et al., 2023; Kim et al., 2024; Pertsch et al., 2025) or generating continuous action chunks through flow matching (Black et al., 2024) and diffusion (Li et al., 2024a; NVIDIA et al., 2025). Recent advances further improve generalization through heterogeneous co-training (Intelligence et al., 2025; Wu et al., 2026; Wang et al., 2026a) and cross-embodiment adaptation (Zheng et al., 2025a). However, many pretrained VLAs lack explicit mechanisms for retaining and using interaction history. This work introduces a lightweight, plug-and-play history module that equips existing pretrained VLAs with memory through fine-tuning.

Memory-Augmented Robotic Policies. History-aware policies address partial observability through recurrent policy heads (Li et al., 2024b) or attention over observation windows (Team et al., 2024). Beyond short windows, full-history encoding has shown benefits in visuomotor policy learning (Zhou et al., 2025; Guan et al., 2026), while evolving latent states enable efficient long-context modeling in diffusion policies (Gui et al., 2026). However, incorporating extensive observation histories directly into pretrained VLA backbones incurs substantial computational and memory costs (Koo et al., 2026). Recent VLA approaches therefore use compact representations, including aggregated moment tokens (Zhou et al., 2026), recurrent memory queries (Li et al., 2026), and compressed historical input tokens (Wang et al., 2026b). Our method combines full-history temporal encoding with a persistent initial-scene spatial reference, providing dual memory without extending the VLM input sequence.

3 METHOD

3.1 OVERVIEW OF FAME-VLA

Problem Formulation. We consider long-horizon robotic manipulation under partial observability, where action selection may depend on the interaction history. At timestep t, conventional pretrained VLA policies predict an action chunk $A_t = (a_t, \ldots, a_{t+H-1})$ from the current or short-window visual context o_t, proprioceptive state s_t, and language instruction ℓ:

$$A_t \sim \pi_\theta(\cdot \mid o_t, s_t, \ell). \tag{1}$$

However, current or recent observations may be insufficient when the appropriate action depends on earlier observations or interactions. We therefore augment the pretrained policy with a history module $\mathcal{M}_\psi$, which encodes the variable-length observation history $\mathcal{H}_t$, spanning the episode start through timestep t, into a compact memory condition:

$$c_t = \mathcal{M}_\psi(\mathcal{H}_t), \qquad A_t \sim \pi_{\theta'}(\cdot \mid o_t, s_t, \ell, c_t), \tag{2}$$

where $\pi_{\theta'}$ denotes the history-augmented policy. This augmentation turns the pretrained VLA into a history-aware policy that incorporates past information into action prediction.

Overview. As illustrated in Figure 2, FAME-VLA augments pretrained VLAs with a lightweight dual-memory module for history-aware action prediction. It combines a full-history temporal memory that captures evolving interaction context through causal encoding with a persistent initial-scene memory that retains spatial details for retrieval based on the current observation (Section 3.2). The resulting memory representation guides action generation through zero-initialized modulation at every action-expert layer, while keeping the vision-language input sequence unchanged (Section 3.3). We jointly train the memory modules and policy-adaptation parameters using the original action-learning objective. An episode-wise training scheme shares a single causal history encoding across sampled timesteps to avoid repeated computation over overlapping prefixes (Section 3.4).

Memory-dependent manipulation requires historical context to resolve ambiguities, yet most pretrained VLAs rely on current or short-window observations. Frame stacking or keyframe selection can incur redundant computation or overlook subtle task-relevant events, motivating a compact encoding of the full observed history. However, compression may attenuate spatial details needed to recall the scene's original configuration, for which the initial observation provides a useful reference (Zhu et al., 2026). To this end, we design a dual-memory representation that combines causal full-history temporal encoding with persistent initial-scene spatial features, jointly providing interaction context and an observation-dependent spatial reference for action prediction.

Full-History Temporal Memory. The temporal memory summarizes the evolving interaction context from the complete observed history. For each primary-camera image I_t, we extract patch features using the pretrained VLA's visual encoder. Let $p_{t,i} \in \mathbb{R}^d$ denote the i-th spatial token mapped into the history representation space. To adaptively capture the importance of different spatial regions for action prediction, we aggregate these tokens through a learned spatial attention pooling mechanism:

$$\alpha_{t,i} = \frac{\exp(w^\top p_{t,i})}{\sum_{j=1}^{N} \exp(w^\top p_{t,j})}, \qquad v_t = \sum_{i=1}^{N} \alpha_{t,i} p_{t,i}. \tag{3}$$

where N is the number of spatial tokens and $w \in \mathbb{R}^d$ is a learnable scoring vector shared across timesteps. Through action supervision, this pooling mechanism learns content-dependent weights that emphasize cues useful for action prediction.

We fuse the visual summary with the proprioceptive state through a projected additive adapter:

$$z_t = \mathrm{LN}\left(v_t + W_s s_t\right). \tag{4}$$

To integrate evidence across the observed history, a temporal encoder $\mathcal{T}_\phi$ processes the resulting sequence:

$$h_{0:t} = \mathcal{T}_\phi(z_{0:t}). \tag{5}$$

We instantiate $\mathcal{T}_\phi$ as a causal Transformer with rotary positional embeddings (RoPE) (Su et al., 2023) to encode temporal order. By attending to the complete observed prefix $z_{0:\tau}$, each output h_τ integrates visual and proprioceptive evidence across time. The current output h_t serves as a temporal memory that summarizes the interaction context available at timestep t.

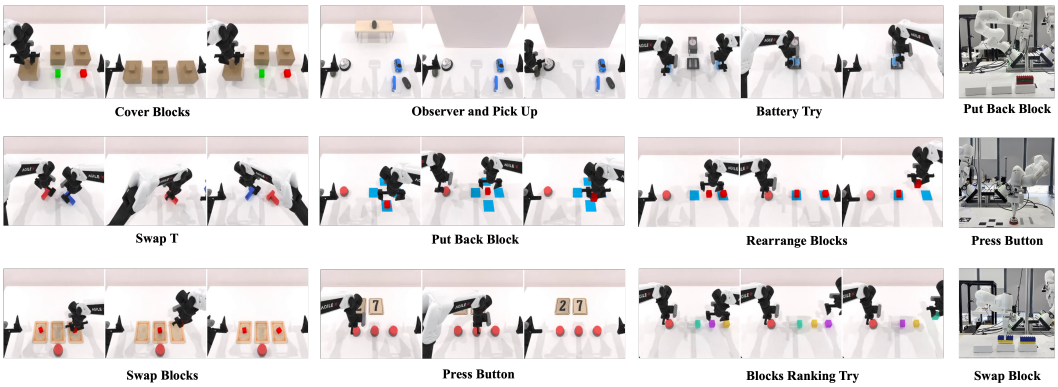


Figure 3: Overview of the evaluation tasks, including nine simulation tasks from RMBench (left) and three real-world tasks (right), all requiring memory of earlier observations or interactions.

3.2 Dual-Memory Representation

Persistent Initial-Scene Memory. The temporal memory h_t provides a compact summary of the observed history, but spatial details from the initial observation may be attenuated by per-frame pooling and temporal encoding. We therefore retain the projected spatial tokens of the first frame as a persistent memory:

$$U_0 = [p_{0,1}; \ldots; p_{0,N}] \in \mathbb{R}^{N \times d}. \tag{6}$$

At timestep t, the current visual summary queries this memory through a cross-attention module:

$$r_t = \mathcal{C}_\psi(v_t, U_0), \tag{7}$$

where v_t provides the query and U_0 provides the keys and values. Although U_0 remains fixed within an episode, the query-dependent attention produces an observation-dependent readout of the initial scene. The integration of r_t with the temporal memory and their injection into the action expert are described in Section 3.3.

3.3 Memory-Modulated Action Expert

Given the dual-memory representation, we seek to incorporate historical context into action generation while retaining the pretrained VLA's visual-semantic capabilities. This context helps resolve visual aliasing, where similar current observations require different actions depending on earlier observations and interactions. To this end, we apply AdaLN-style modulation (Peebles & Xie, 2023) at every action-expert layer without extending the vision-language input sequence. The modulation mappings are zero-initialized to preserve the pretrained computation at initialization.

At each decision timestep, we concatenate the temporal memory h_t with the observation-dependent readout r_t of the persistent initial-scene memory to form a joint condition $c_t = [h_t; r_t]$. For each layer k and sublayer $s \in \{\mathrm{attn}, \mathrm{ffn}\}$, a lightweight modulation mapping produces feature-wise scale offsets and shifts:

$$[\gamma^s_{k,t}; \beta^s_{k,t}] = \mathcal{M}^s_k(c_t). \tag{8}$$

These parameters modulate the normalized input to the corresponding sublayer:

$$\widehat{x}^s_{k,t} = \mathrm{Norm}^s_k(x^s_{k,t}) \odot (1 + \gamma^s_{k,t}) + \beta^s_{k,t}, \tag{9}$$

where $x^s_{k,t}$ denotes the sublayer input and Norm^s_k is the existing normalization operator.

We initialize the modulation mappings to output $\gamma^s_{k,t} = \beta^s_{k,t} = 0$, so the added memory conditioning leaves the original normalized activations unchanged at initialization. This design incorporates both forms of memory into action generation without extending the vision-language input sequence.

Table 1: Main results on RMBench. Tasks are grouped by memory complexity into single-memory ($M(1)$) and multi-memory ($M(n)$) tasks. **Bold** indicates the best result in each column.

	$M(1)$ tasks					$M(n)$ tasks				Average		
Method	Observe and Pick Up	Rearrange Blocks	Put Back Block	Swap Blocks	Swap T	Battery Try	Blocks Ranking Try	Cover Blocks	Press Button	$M(1)$	$M(n)$	Overall
Standard policy baselines												
DP(Chi et al., 2024)	1%	0%	0%	11%	20%	10%	10%	0%	0%	6.40%	5.00%	5.78%
ACT(Zhao et al., 2023)	1%	29%	0%	2%	2%	19%	0%	0%	0%	6.80%	4.75%	5.89%
$\pi_{0.5}$(Intelligence et al., 2025)	**9%**	13%	11%	24%	15%	16%	6%	0%	0%	14.40%	5.50%	10.44%
X-VLA(Zheng et al., 2025a)	**9%**	13%	18%	16%	3%	26%	1%	2%	0%	11.80%	7.25%	9.78%
Memory-augmented baselines												
MTIL(Zhou et al., 2025)	1%	74%	37%	2%	35%	13%	41%	0%	0%	29.80%	13.50%	22.56%
CycleManip(Wei et al., 2026)	1%	0%	0%	1%	88%	14%	84%	10%	0%	18.00%	27.00%	22.00%
MemER(Sridhar et al., 2025)	7%	17%	0%	14%	7%	27%	0%	6%	0%	9.00%	8.25%	8.67%
Mem-0(Chen et al., 2026)	4%	89%	90%	67%	14%	28%	18%	68%	0%	52.80%	28.50%	42.00%
MemoryVLA(Shi et al., 2026)	2%	53%	81%	76%	9%	33%	53%	69%	0%	44.20%	38.75%	41.78%
HAMLET(Koo et al., 2026)	4%	22%	23%	15%	20%	29%	83%	13%	0%	16.80%	31.25%	23.22%
Backbone-only baselines (without task memory)												
StarVLA(Community, 2026)	0%	0%	0%	0%	0%	14%	37%	0%	0%	0.00%	12.75%	5.67%
π_0(Black et al., 2024)	8%	10%	16%	8%	13%	9%	0%	0%	0%	11.00%	2.25%	7.11%
Our memory framework with interchangeable backbones												
FAME-VLA (StarVLA)	2%	**100%**	**100%**	96%	95%	34%	28%	46%	**4%**	**78.60%**	28.00%	56.11%
FAME-VLA (π_0)	1%	97%	81%	**98%**	**96%**	**40%**	**91%**	**79%**	1%	74.60%	**52.75%**	**64.89%**

3.4 Episode-Wise Training with Shared History Encoding

Incorporating full observation histories into VLA training introduces redundant computation when overlapping prefixes are encoded independently for different supervised timesteps. To reduce this overhead, we adopt episode-wise training with shared history encoding, allowing multiple action predictions to share a single causal encoding pass. The memory modules and pretrained models are jointly optimized in a single stage under the original action-learning objective, allowing action supervision to shape task-relevant memory representations.

Shared History Encoding. Since the visual encoder is frozen, we precompute and cache its spatial features for reuse across training updates. For each sampled episode of length T, the trainable spatial pooling and proprioceptive fusion modules construct the per-step representations $z_{0:T-1}$. A single forward pass of the causal history encoder computes all temporal memories $h_{0:T-1}$ in parallel. Causal masking restricts each h_t to $z_{0:t}$, allowing the entire episode to be processed together while preserving the information available at each decision timestep.

Joint Optimization. We sample decision timesteps $\mathcal{S}$ across the episode, gather their temporal memories, and combine them with the corresponding initial-scene readouts to obtain $c_t = [h_t; r_t]$. These conditions, together with the associated observations and action targets, form the policy training batch. Multiple decisions thus share one history encoding, avoiding repeated computation over overlapping prefixes. Policy evaluation for action supervision is restricted to the sampled timesteps, while each decision retains access to its complete causal history.

We retain the underlying VLA's action-learning objective and average it over the sampled timesteps:

$$\mathcal{L}_{\text{episode}} = \frac{1}{|\mathcal{S}|} \sum_{t \in \mathcal{S}} \mathcal{L}_{\text{act}}\big(A_t; o_t, s_t, \ell, c_t\big). \tag{10}$$

where $\mathcal{L}_{\text{act}}$ denotes the action-prediction objective of the pretrained VLA. We jointly optimize the proposed modules and policy-adaptation parameters to learn task-relevant history representations through action supervision.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks and Tasks. We evaluate FAME-VLA on memory-dependent manipulation tasks in both simulation and the real world. For simulation, we use all nine tasks from RMBench (Chen et al., 2026), which categorizes tasks by Task Memory Complexity (TMC) according to the amount of task-relevant history required for decision making. Specifically, $M(1)$ tasks require information from a single key past observation, whereas $M(n)$ tasks require multiple task-relevant observations across the interaction history.

For real-world evaluation, we use a single-arm Franka robotic platform equipped with an Intel RealSense D435 camera for the external view and an Intel RealSense D405 camera mounted on the wrist. We design three tabletop tasks that require tracking task progress or remembering earlier scene configurations: (i) *Press Button*, where the robot presses the button three times before returning to the initial position; (ii) *Put Back Block*, where the robot returns a previously moved block to its original location; and (iii) *Swap Blocks*, where the robot swaps the positions of two visually identical blocks using an intermediate buffer location. Detailed task descriptions are provided in Appendix A.

Baselines. We compare FAME-VLA with memory-free policies, including DP (Chi et al., 2024), ACT (Zhao et al., 2023), $\pi_{0.5}$ (Black et al., 2024), and X-VLA (Zheng et al., 2025a), as well as representative memory-augmented approaches, including MemER (Sridhar et al., 2025), Mem-0 (Chen et al., 2026), MemoryVLA (Shi et al., 2026), HAMLET (Koo et al., 2026), MTIL (Zhou et al., 2025) and CycleManip (Wei et al., 2026). To isolate the effect of our memory framework and assess its applicability across VLA architectures, we further instantiate FAME-VLA with two pretrained backbones, π_0 and StarVLA, and compare each against its backbone-only counterpart.

Evaluation Protocol. We use task success rate as the primary evaluation metric, computed as the proportion of trials that satisfy the task-specific success criteria. On RMBench, we evaluate each method over 100 rollout episodes per task. For real-world evaluation, we conduct 20 trials per task and report the corresponding success rates. All methods are evaluated using the same success criteria and evaluation configurations for each task to ensure consistent comparisons.

Implementation Details. We use 50 demonstrations per task for post-training on both RMBench and the real-world tasks. Training is conducted on NVIDIA A100. We use the episode-wise training with shared history encoding described in Sec. 3.4, while retaining the original action objective and backbone-specific optimization setup for each pretrained VLA. Additional architectural configurations, training hyperparameters, and implementation details are provided in Appendix A.

4.2 SIMULATION EXPERIMENTS

Main Results. We evaluate our plug-in memory module on RMBench with StarVLA and π_0 as backbones. As shown in Table 1, incorporating FAME-VLA raises the overall success rate from 5.67% to 56.11% for StarVLA and from 7.11% to 64.89% for π_0, supporting its effectiveness across both architectures. The π_0 variant further exceeds the strongest existing memory-augmented method by 22.89 percentage points. Across memory-complexity groups, the StarVLA and π_0 variants achieve the highest average success rates on $M(1)$ and $M(n)$ tasks, respectively, indicating effectiveness under different memory requirements.

Ablation Study. We first examine whether both memory branches contribute to memory-dependent decision making (Table 2). Removing full-history temporal memory reduces overall success from 64.89% to 13.89%, indicating that initial-scene information alone cannot adequately capture task context accumulated throughout execution. Removing persistent initial-scene memory also lowers success to 50.67%, suggesting that explicit access to initial spatial details provides useful information beyond the compact temporal representation. Together, these results support the complementary roles of the two memory branches.

We further ablate how memory representations are constructed and incorporated into the action expert. Replacing learnable spatial pooling with mean pooling reduces overall success to 52.00%. This supports learning spatial aggregation jointly with the policy, suggesting that action supervision helps

Table 2: Ablation results on RMBench using π_0 as the backbone. Each variant removes or replaces one component of FAME-VLA. **Bold** indicates the best result in each column.

Method	$M(1)$	$M(n)$	Overall
π_0	11.00%	2.25%	7.11%
Ablations of the proposed memory framework			
w/o Full-History Temporal Memory	22.60%	3.00%	13.89%
w/o Persistent Initial-Scene Memory	65.40%	32.25%	50.67%
w/ Mean Spatial Pooling	62.00%	39.50%	52.00%
w/ History Tokens as Action-Expert Prefix	67.60%	28.50%	50.22%
FAME-VLA (π_0)	**74.60%**	**52.75%**	**64.89%**

preserve task-relevant information in compact per-frame representations for full-history encoding. Beyond representation construction, replacing layer-wise adaptive modulation with a history-token prefix lowers overall success to 50.22%. The decrease is larger on $M(n)$ tasks (24.25 percentage points) than on $M(1)$ tasks (7.00 percentage points), suggesting that layer-wise modulation is particularly beneficial when action generation requires information from multiple past observations.

Analysis of History Length. We investigate how history length affects performance on three memory-dependent tasks (Table 3). Full-history conditioning achieves the best average success rate, matching or exceeding all finite-window variants on each task. More broadly, average success consistently improves as the history window expands, with full history providing further gains. This trend suggests that task-relevant information can extend beyond a short observation window, making access to earlier observations valuable for memory-dependent decision making. Together, these results support the benefits of modeling the complete observed history.

These performance gains come with a modest increase in inference cost. Under full-history conditioning, FAME-VLA maintains end-to-end P95 latency close to that of finite-window variants, with approximately 18.0% overhead over the π_0 backbone. These results show that extending temporal memory to the complete observed history improves task performance with little additional end-to-end latency compared with finite-window conditioning.

Table 3: Effect of temporal memory span on success rates and inference latency. W denotes the temporal memory window size. Full uses the complete observed history. Latency is reported at the 95th percentile (P95). Average is computed over the three tasks. **Bold** indicates the best success rate in each column, including ties.

	Success rate ↑				P95 latency ↓	
Setting	Put Back Block	Swap T	Cover Blocks	Average	History encoder	Full policy
π_0	16%	13%	0%	9.67%	–	103.71 ms
$W = 8$	62%	53%	1%	38.67%	2.71 ms	118.68 ms
$W = 16$	69%	46%	22%	45.67%	3.13 ms	120.65 ms
$W = 32$	**81%**	76%	25%	60.67%	2.89 ms	121.52 ms
Full	**81%**	**96%**	**79%**	**85.33%**	4.78 ms	122.37 ms

Efficiency of Episode-Wise Training. We investigate whether episode-wise history sharing enables more efficient training as the number of supervised timestamps per episode increases. As shown in Table 4, the benefit of sharing grows with $|\mathcal{S}|$, as more supervised timestamps entail more redundant computation over overlapping prefixes. By reusing a single causal encoding, our strategy keeps iteration time nearly constant across the evaluated settings, achieving a $4.13\times$ speedup and reducing peak GPU memory from 8.39 to 6.48 GiB at $|\mathcal{S}| = 32$. This allows denser supervision within each episode without repeatedly paying the cost of full-history encoding.

Table 4: Training efficiency of independent prefix encoding (Indep. Prefix) and shared episode-wise encoding (Shared Episode), using identical cached visual features and supervised timesteps. $|\mathcal{S}|$ denotes the number of supervised timesteps per episode. Throughput measures supervised timesteps processed per second. Speedup is based on median (P50) iteration time.

$\lvert\mathcal{S}\rvert$	Strategy	Time / iteration ↓ P50 / P95 (ms)	Throughput ↑ (timesteps/s)	Peak allocated ↓ (GiB)	Speedup ↑
4	Indep. Prefix	9.71 / 10.59	412	6.66	1.00×
	Shared Episode	**8.55 / 8.77**	**468**	**6.48**	**1.14×**
16	Indep. Prefix	22.86 / 23.05	700	7.43	1.00×
	Shared Episode	**8.60 / 9.05**	**1,861**	**6.48**	**2.66×**
32	Indep. Prefix	35.41 / 36.03	904	8.39	1.00×
	Shared Episode	**8.57 / 8.67**	**3,734**	**6.48**	**4.13×**

4.3 Real-World Experiments

We evaluate FAME-VLA on three real-world manipulation tasks that require tracking task progress or recalling prior scene configurations, as shown in Table 5. FAME-VLA substantially outperforms the π_0 baseline across all tasks, increasing the average success rate from 10.00% to 71.67%. The gains are particularly pronounced on Press Button, where FAME-VLA improves the success rate from 10% to 90%. These results demonstrate the effectiveness of our memory framework for real-world memory-dependent manipulation, where decisions depend on past events and initial scene configurations beyond the current observation. To better understand the remaining challenges, we provide a failure-mode analysis for each task in Appendix C and discuss limitations in Section 6.

Table 5: Success rates (%) on real-world memory-dependent manipulation tasks using π_0 as the backbone. **Bold** indicates the best result in each column.

Method	Press Button	Swap Blocks	Put Back Block	Average
π_0	10	5	15	10.00
FAME-VLA(π_0)	**90**	**55**	**70**	**71.67**

5 Conclusion

In this paper, we introduce FAME-VLA, a lightweight full-history adapter for pretrained vision-language-action models. Our results show that combining full-history context with initial-scene details improves memory-dependent manipulation across simulated and real-world tasks. The proposed dual-memory representation and zero-initialized layer-wise modulation integrate complementary historical information into action generation, while trajectory-wise history sharing improves training efficiency by avoiding repeated encoding of overlapping observation prefixes.

6 Limitations

Our method has two main limitations. First, the memory module is learned through action supervision on downstream demonstrations, and its generalization to unseen tasks, longer executions, and novel task compositions remains insufficiently explored. An important direction is to investigate whether the learned history representations capture reusable temporal dependencies beyond the training tasks. Second, the persistent initial-scene memory relies on the first observation as a privileged source of spatial detail. This reference may be less informative when relevant objects are initially occluded or introduced later during execution. Extending this branch to selectively retain and update spatial memories from subsequent observations could broaden its applicability.

AI USE STATEMENT

We used generative AI tools to assist with implementing the code and polishing the writing of the manuscript. All AI-assisted code and text were reviewed, and the implementation was tested by the authors. We take responsibility for the final content of this work, including all text, claims, code, data, and other artifacts.

ETHICS STATEMENT

We have considered the issues covered by the ICLR Code of Ethics and do not identify any ethical concerns raised by this work.

REPRODUCIBILITY STATEMENT

We will release the complete source code, training and evaluation scripts, and trained model checkpoints. The main paper and appendix document the policy implementations, training configurations, and evaluation protocols.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael 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, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control, November 2024. URL http://arxiv.org/abs/2410.24164. arXiv:2410.24164 [cs].

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-Time Execution of Action Chunking Flow Policies, December 2025. URL http://arxiv.org/abs/2506.07339. arXiv:2506.07339 [cs].

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

Tianxing Chen, Yuran Wang, Mingleyang Li, Yan Qin, Hao Shi, Zixuan Li, Yifan Hu, Yingsheng Zhang, Kaixuan Wang, Yue Chen, Hongcheng Wang, Renjing Xu, Ruihai Wu, Yao Mu, Yaodong Yang, Hao Dong, and Ping Luo. RMBench: Memory-Dependent Robotic Manipulation Benchmark with Insights into Policy Design, March 2026. URL http://arxiv.org/abs/2603.01229. arXiv:2603.01229 [cs.RO] version: 2.

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, 2026. URL https://arxiv.org/abs/2606.12497. Version Number: 1.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion, March 2024. URL http://arxiv.org/abs/2303.04137. arXiv:2303.04137 [cs].

StarVLA Community. StarVLA: A Lego-like Codebase for Vision-Language-Action Model Developing, April 2026. URL http://arxiv.org/abs/2604.05014. arXiv:2604.05014 [cs.RO].

Zhiyuan Guan, Jianshu Hu, Han Fang, Yunpeng Jiang, Yize Huang, Shujia Li, Xiao Li, and Yutong Ban. Dssp: Diffusion state space policy with full-history encoding, 2026. URL https://arxiv.org/abs/2605.14598.

Youqiang Gui, Yuxuan Zhou, Shen Cheng, Xinyang Yuan, Haoqiang Fan, Peng Cheng, and Shuaicheng Liu. SeedPolicy: Horizon Scaling via Self-Evolving Diffusion Policy for Robot Manipulation, March 2026. URL http://arxiv.org/abs/2603.05117. arXiv:2603.05117 [cs].

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael 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, April 2025. URL http://arxiv.org/abs/2504.16054. arXiv:2504.16054 [cs].

Huiwon Jang, Sihyun Yu, Heeseung Kwon, Hojin Jeon, Younggyo Seo, and Jinwoo Shin. ContextVLA: Vision-Language-Action Model with Amortized Multi-Frame Context, October 2025. URL http://arxiv.org/abs/2510.04246. arXiv:2510.04246 [cs].

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag 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, September 2024. URL http://arxiv.org/abs/2406.09246. arXiv:2406.09246 [cs].

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, April 2026. URL http://arxiv.org/abs/2510.00695. arXiv:2510.00695 [cs].

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H. Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior Generation with Latent Actions, June 2024. URL http://arxiv.org/abs/2403.03181. arXiv:2403.03181 [cs].

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, 2026. 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. CronusVLA: Towards Efficient and Robust Manipulation via Multi-Frame Vision-Language-Action Modeling, October 2025. URL http://arxiv.org/abs/2506.19816. arXiv:2506.19816 [cs].

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, 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, 2024b. URL https://arxiv.org/abs/2311.01378.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a Diffusion Foundation Model for Bimanual Manipulation, March 2025. URL http://arxiv.org/abs/2410.07864. arXiv:2410.07864 [cs].

Max Sobol Mark, Jacky Liang, Maria Attarian, Chuyuan Fu, Debidatta Dwibedi, Dhruv Shah, and Aviral Kumar. BPP: Long-Context Robot Imitation Learning by Focusing on Key History Frames, February 2026. URL http://arxiv.org/abs/2602.15010. arXiv:2602.15010 [cs].

NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers, March 2023. URL http://arxiv.org/abs/2212.09748. arXiv:2212.09748 [cs].

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, January 2025. URL http://arxiv.org/abs/2501.09747. arXiv:2501.09747 [cs].

Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior Transformers: Cloning k modes with one stone, October 2022. URL http://arxiv.org/abs/2206.11251. arXiv:2206.11251 [cs].

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, January 2026. URL http://arxiv.org/abs/2508.19236. arXiv:2508.19236 [cs.RO].

Ajay Sridhar, Jennifer Pan, Satvik Sharma, and Chelsea Finn. MemER: Scaling Up Memory for Robot Control via Experience Retrieval, October 2025. URL http://arxiv.org/abs/2510.20328. arXiv:2510.20328 [cs.RO].

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy, 2024. URL https://arxiv.org/abs/2405.12213.

Marcel Torne, Karl Pertsch, Homer Walke, Kyle Vedder, Suraj Nair, Brian Ichter, Allen Z Ren, Haohuan Wang, Jiaming Tang, Kyle Stachowicz, Karan Dhabalia, Michael Equi, Quan Vuong, Jost Tobias Springenberg, Sergey Levine, Chelsea Finn, and Danny Driess. MEM: Multi-Scale Embodied Memory for Vision Language Action Models.

Qiuyue Wang, Mingsheng Li, Jian Guan, Jinhui Ye, Sicheng Xie, Yitao Liu, Junhao Chen, Zhixuan Liang, Jie Zhang, Xintong Hu, Xuhong Huang, Pei Lin, Junyang Lin, Dayiheng Liu, Shuai Bai, Jingren Zhou, Jiazhao Zhang, Haoqi Yuan, Gengze Zhou, Hang Yin, Ye Wang, Yiyang Huang, Zixing Lei, Wujian Peng, Delin Chen, Yingming Zheng, Jingyang Fan, Xianwei Zhuang, Xin Zhou, Haoyang Li, Anzhe Chen, Tong Zhang, Xuejing Liu, Yuchong Sun, Ruizhe Chen, Zhaohai Li, Chenxu Lü, Zhibo Yang, Tao Yu, and Xionghui Chen. Qwen-VLA: Unifying Vision-Language-Action Modeling across Tasks, Environments, and Robot Embodiments, June 2026a. URL http://arxiv.org/abs/2605.30280. arXiv:2605.30280 [cs.RO].

Ziye Wang, Modi Shi, Chaojun Ni, Jiazhi Yang, Mengdi Li, Zhizhong Su, Tianwei Lin, and Hongyang Li. NativeMEM: Native Memory Compression for Long-Horizon Robotic Manipulation, July 2026b. URL https://arxiv.org/abs/2607.06678. Version Number: 1.

Yi-Lin Wei, Haoran Liao, Yuhao Lin, Pengyue Wang, Zhizhao Liang, Guiliang Liu, and Wei-Shi Zheng. CycleManip: Enabling Cyclic Task Manipulation via Effective Historical Perception and Understanding, March 2026. URL http://arxiv.org/abs/2512.01022. arXiv:2512.01022 [cs.RO].

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, Yiyu Ren, Kejia Zhang, Hui Yu, Jingmei Zhao, Shuai Zhou, Zhenqi Qiu, Houlong Xiong, Ziyu Wang, Zechen Wang, Ran Cheng, Yong-Lu Li, Yongtao Huang, Xing Zhu, Yujun Shen, and Kecheng Zheng. A pragmatic vla foundation model, 2026. URL https://arxiv.org/abs/2601.18692.

Ganlin Yang, Zhangzheng Tu, Yuqiang Yang, Sitong Mao, Junyi Dong, Tianxing Chen, Jiaqi Peng, Jing Xiong, Jiafei Cao, Jifeng Dai, Wengang Zhou, Yao Mu, and Tai Wang. EventVLA: Event-Driven Visual Evidence Memory for Long-Horizon Vision-Language-Action Policies, June 2026. URL http://arxiv.org/abs/2606.20092. arXiv:2606.20092 [cs.CV].

Yihan Zeng, Minghao Ye, Yiyuan Chen, Yide Shentu, Philipp Wu, Zike Yan, and Zhongyu Li. Kemo: Event-driven keyframe memory for long-horizon robot manipulation with vla policies, 2026. URL https://arxiv.org/abs/2606.23589.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware, April 2023. URL http://arxiv.org/abs/2304.13705. arXiv:2304.13705 [cs].

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-Prompted Transformer as Scalable Cross-Embodiment Vision-Language-Action Model, October 2025a. URL http://arxiv.org/abs/2510.10274. arXiv:2510.10274 [cs.RO].

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daume Iii, Andrey Kolobov, Furong Huang, and Jianwei Yang. TRACEVLA: VISUAL TRACE PROMPTING ENHANCES SPATIAL-TEMPORAL AWARENESS FOR GENERALIST ROBOTIC POLICIES. 2025b.

Yijing Zhou, Qiwei Liang, Sitong Zhuang, Jiaxi Li, Xianpeng Wang, Boyang Cai, Yunyang Mo, and Renjing Xu. Action-Effect Memory Pretraining for Robot Manipulation, June 2026. URL http://arxiv.org/abs/2606.12499. arXiv:2606.12499 [cs.RO].

Yulin Zhou, Yuankai Lin, Fanzhe Peng, Jiahui Chen, Kaiji Huang, Hua Yang, and Zhouping Yin. MTIL: Encoding Full History With Mamba for Temporal Imitation Learning. *IEEE Robotics and Automation Letters*, 10(11):11761–11767, November 2025. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2025.3615520. URL https://ieeexplore.ieee.org/document/11184145/.

Juan Zhu, Zhanying Shao, Xiaoqi Li, Ethan Morgan, Jiadong Xu, Hongwei Fan, and Hao Dong. Anchorvla4d: an anchor-based spatial-temporal vision-language-action model for robotic manipulation, 2026. URL https://arxiv.org/abs/2603.12730.

A Appendix

A.1 Implementation Details

We instantiate FAME-VLA with π_0 and StarVLA using the same dual-memory design. For both backbones, full-history temporal memory is constructed through learnable spatial pooling and a causal Transformer, while persistent initial-scene memory retrieves initial visual details using current visual features as queries. The two memory representations jointly condition the action expert through zero-initialized, layer-wise adaptive modulation, without extending the VLM input sequence.

During training, we freeze the visual encoder and adapt the pretrained VLA through LoRA, jointly optimizing the LoRA parameters and the newly introduced memory and modulation modules under each backbone's original action prediction objective. We adopt trajectory-wise training with shared history encoding and sample $K = 32$ temporally stratified supervision timesteps per trajectory, using an action chunk length of $H = 50$ and 50 expert demonstrations per task. Detailed architectural configurations and training hyperparameters for π_0 and StarVLA are provided in Tables 6 and 7, respectively.

Table 6: Hyperparameters used with the π_0 backbone.

Hyperparameter	Value
Demonstrations per Task	50
Action Chunk Length (H)	50
Anchor Samples per Episode (K)	32
History Encoder	Causal Transformer
History Encoder Layers	2
History Encoder Attention Heads	8
History Encoder Hidden Dimension	512
History Encoder MLP Dimension	2048
Positional Encoding	RoPE, $\theta = 10{,}000$
Initial-Scene Cross-Attention Layers	2
Memory Condition Dimension	1024
Action Objective	Flow-matching MSE
Optimizer	AdamW
Optimizer Betas (β_1, β_2)	$(0.9, 0.95)$
Learning Rate	2.5×10^{-5}
Minimum Learning Rate	2.5×10^{-6}
Weight Decay	1.0×10^{-10}
Learning Rate Schedule	Cosine decay with warmup
Warmup Steps	1,000
Training Steps	30,000
Gradient Clipping Norm	1.0

Table 7: Hyperparameters used with the StarVLA backbone.

Hyperparameter	Value
Demonstrations per Task	50
Action Chunk Length (H)	50
Anchor Samples per Episode (K)	32
History Encoder	Causal Transformer
History Encoder Layers	2
History Encoder Attention Heads	8
History Encoder Hidden Dimension	512
Initial-Scene Cross-Attention Layers	2
Memory Condition Dimension	1024
Action Head	2-block MLP-ResNet
Action Objective	L1 regression
Optimizer	AdamW
Optimizer Betas (β_1, β_2)	$(0.9, 0.95)$
VLM Learning Rate	1.0×10^{-5}
Action Head Learning Rate	1.0×10^{-4}
Minimum Learning Rate	5.0×10^{-7}
Weight Decay	1.0×10^{-8}
Learning Rate Schedule	Cosine decay with warmup
Warmup Steps	2,000
Training Steps	30,000

B Real-World Task Protocol and Training Details

B.1 Task Definitions and Demonstrations

We evaluate π_0 and π_0 augmented with FAME-VLA on three memory-dependent tabletop manipulation tasks: Press Button, Put Back Block, and Swap Blocks. Each policy controls a single 7-DoF Franka arm and its gripper. Table 8 summarizes the language prompts used for training.

Table 8: Language prompts for the real-world tasks.

Task name	Prompt
Press Button	*"Press the button three times, then return to the initial position."*
Put Back Block	*"Move the block from the left or right side to the center, then return it to its original position."*
Swap Blocks	*"Swap the left and right blocks by moving them through the center."*

Press Button. The robot presses a button three times and then returns to its initial configuration. This task requires tracking the number of completed presses, since the button returns to a similar visual state after each press. The language prompt specifies both the three presses and the subsequent return to the initial configuration.

Put Back Block. The robot moves a block from either the left or right location to the center and then returns it to its original location. Once the block reaches the center, its current position alone does not reveal which side it came from. The policy must therefore retain information about the initial configuration to select the correct return location.

Swap Blocks. The robot exchanges two visually identical blocks between the left and right locations using the center as a temporary buffer. Completing the exchange requires tracking the movement history and manipulation stage, because the blocks' appearance alone does not distinguish their identities or indicate whether the exchange has been completed.

C Failure-Mode Analysis on Real-World Tasks

We analyze the failure modes of the three real-world tasks to understand how memory contributes to the improvements of FAME-VLA and what challenges remain.

Press Button. This task requires the robot to press the button three times and then stop. Since the robot returns to a similar pose after each press, the current observation provides little information about the number of completed presses. Without memory, π_0 tends to continue pressing beyond the required count instead of stopping at the correct stage. By incorporating full-history context, FAME-VLA can track the number of completed presses and determine when to begin the return motion.

Put Back Block. This task requires the robot to press the button three times and then return to its initial configuration. Once the block reaches the center, the current observation no longer reveals which side it came from. The π_0 baseline often hesitates at this intermediate stage or places the block on the opposite side, failing to restore the initial configuration. With access to initial-scene memory and historical context, FAME-VLA can identify the original position and guide the subsequent return movement.

Swap Blocks. This task requires multiple pick-and-place operations to swap two blocks through an intermediate buffer location. The longer manipulation sequence creates more opportunities for grasping and placement errors, which remain a source of failure for FAME-VLA. The buffer stage also introduces visual aliasing: similar current observations can require different subsequent movements depending on the preceding sequence. Without historical context, π_0 often hesitates at this stage, interrupting task progress. Our memory framework helps disambiguate the buffer state and determine the next movement, while successful completion still depends on accurate execution of each pick-and-place operation.