KEEP EVERYTHING, READ ONE LATENT: VERBATIM EPISODE MEMORY FOR WORLD-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World-action models (WAMs) typically act from a bounded window of recent observations, which makes their decisions effectively Markovian. We show that a WAM becomes history-dependent with verbatim episode memory (VEM). Every observation, from the demonstration and from execution alike, stays in context as its encoded video latent. After temporal sampling and native VAE encoding, no additional memory compression, selection, or eviction is applied. Training uses expert actions and future video targets, without memory annotations or subgoal supervision. The video expert integrates the causal history into the current latent, while the action expert reads only this fixed-size, history-conditioned state. VEM achieves state-of-the-art overall performance on RoboMME, with a 62.5% average success rate across 16 tasks. Placing the demonstration and execution history on one stream elicits strong imitation capabilities, yielding a 78.2% success rate on the Imitation Suite. This outperforms policies conditioned on ground-truth subgoals on both our backbone (49.5%) and a language-conditioned VLA (64.0%). In a retrieval test, a single initial observation still steers behavior after more than twice the length of the longest training episode of irrelevant history.

1 INTRODUCTION

Many manipulation tasks cannot be solved from the current observation alone. The relevant evidence may lie many seconds in the past, and which part of the past matters is often unknown until the moment of acting. World-action models (WAMs) (Li et al., 2025; Kim et al., 2026; Li et al., 2026; Ye et al., 2026) are a natural candidate for such tasks: they build policies on pretrained video generators, whose backbones are trained to model long visual sequences. Yet most WAMs condition on a bounded window of recent observations, which makes their decisions effectively Markovian in that window.

Existing approaches to memory in robot policies decide in advance what part of the past the policy will see. Symbolic methods have a high-level model track keyframes or summarize the episode into language subgoals (Sridhar et al., 2026; Torne et al., 2026; Gemini Robotics Team et al., 2025); perceptual methods keep a selected or budgeted set of past visual features (Jang et al., 2025; Koo et al., 2026; Shi et al., 2026; Fang et al., 2025); recurrent methods compress history into a fixed-size state (Mandlekar et al., 2021; Liu et al., 2024; Zhou et al., 2025; Jiang et al., 2026); and long-horizon WAMs combine recent frames with anchor frames and gist tokens chosen by event boundaries (Yang et al., 2026). Each adds a memory-specific component—a selection rule, a budget, a summarizer, or a recurrent update—and whatever that component discards is unavailable to the policy later. On the RoboMME benchmark (Dai et al., 2026), which compares these families on a shared backbone, the strongest memory policy reaches 44.5% average success, and even ground-truth subgoals leave imitation tasks far from solved.

We take the opposite approach: keep everything. *Verbatim episode memory* (VEM) places the entire episode, demonstration and execution alike, on a single causal stream of video latents. After temporal sampling and native VAE encoding, all resulting episode latents are retained without additional memory compression, selection, or eviction. No memory module, memory annotations, or subgoal supervision is added; training uses expert actions and future video targets. The video expert attends causally over this stream and integrates it into the current latent f_0; the action expert reads only f_0, a fixed-size, history-conditioned predictive state. Because the stream is causal, past latents never

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.

change once encoded, so deployment reuses a key–value cache. Action-decoding cost stays fixed, and KV caching yields favorable per-step latency over the tested history lengths (Figure 2).

This simple design is effective. On RoboMME, VEM reaches 62.5% average success against 44.5% for the strongest prior memory policy, and improves over a fixed-budget context on the same backbone by 9.2 points (§4.2). In a controlled retrieval test, a single initial observation still steers behavior after 65 latents of irrelevant history, more than twice the longest training episode for that task, and a clear advantage remains at 129 latents, beyond the longest episode in the whole training set (§4.3). On imitation tasks, placing the demonstration and execution on one stream is sufficient to elicit strong imitation, with no demonstration encoder, alignment module, or demonstration-specific supervision: success rises from 9.0% without history to 78.2%, exceeding the success rates of oracle-subgoal policies on our backbone (49.5%) and on a language-conditioned VLA (64.0%), with the largest gaps on tasks defined by trajectory shape and contact (§4.4). Finally, we detect no statistically significant difference between direct visual-history access and reading f_0 alone in the fixed-budget ablation (§4.5), suggesting the video expert itself performs most of the temporal integration needed for action, and exposing the action expert to the visual history provides little additional benefit.

In summary, our contributions are:

- We introduce VEM, which turns the full causal visual episode into the memory of a WAM, with no additional memory selection, compression, or retrieval module after native video encoding.
- VEM supports long-range behavioral retrieval far beyond the task training horizon.
- We separate history processing from action generation: the video expert integrates the full episode history into the current latent, and the action expert reads only this fixed-size representation to choose actions.
- One stream of demonstration and execution can elicit strong imitation, outperforming ground-truth grounded subgoals, especially on trajectory-rich tasks.

2 Related Work

Memory for robot policies. Most vision-language-action (VLA) policies act on the current observation alone (Physical Intelligence et al., 2025), and a growing line of work adds memory for non-Markovian tasks. Symbolic approaches move history into language or other abstractions: a high-level VLM tracks keyframes or summarizes the episode and emits subgoals for a low-level policy (Sridhar et al., 2026; Gemini Robotics Team et al., 2025; Torne et al., 2026), or tracked points serve as an object-centric state (Chen et al., 2026). Perceptual approaches keep visual features of the past as extra context tokens (Jang et al., 2025; Koo et al., 2026), as a memory bank queried at every step (Shi et al., 2026; Fang et al., 2025; Qu et al., 2026), or as reused prefix keys and values (Sun et al., 2026). Recurrent approaches compress history into a fixed-size state (Mandlekar et al., 2021; Liu et al., 2024; Zhou et al., 2025; Jiang et al., 2026), and long-context policies can be regularized to retain the past (Torne et al., 2025). Dai et al. (2026) compare these families on a shared $\pi_{0.5}$ backbone under a fixed memory budget of 512 tokens. All of them add a memory-specific component, such as a selection rule, a budget, a summarizer, or a recurrent state, that decides which part of the past the policy sees.

World-action models. World-action models (WAMs) build policies on pretrained video generators and train them to predict future frames together with actions (Li et al., 2025; Kim et al., 2026; Li et al., 2026; Ye et al., 2026). Recent work finds that their benefit comes mainly from video co-training, so future generation can be skipped at test time (Yuan et al., 2026), and that the action module can be far shallower than the video backbone (Ma et al., 2026). Most WAMs condition on a bounded window of recent observations, and those that keep longer histories pay a cost that grows with the history (Yang et al., 2026). MemoryWAM (Yang et al., 2026) therefore combines recent frames, event-boundary anchor frames, and gist tokens that summarize long-range history.

Imitation from demonstrations and memory benchmarks. Performing a task from a demonstration given at test time has long been studied as one-shot imitation (Duan et al., 2017). RoboTTT (Jiang et al., 2026) scales visuomotor context to 8K timesteps by compressing history into fast weights, which enables in-context imitation from human video. Memory benchmarks for

manipulation include MemoryBench (Fang et al., 2025), which targets spatial recall, MIKASA-Robo (Cherepanov et al., 2025), RoboMemArena (Lei et al., 2026), and RoboMME (Dai et al., 2026). We evaluate on RoboMME because every task requires history, its Imitation suite can be solved only from the demonstration, and it reports baselines from every memory family on one backbone.

3 METHOD

Figure 1 summarizes the visual information path through VEM.

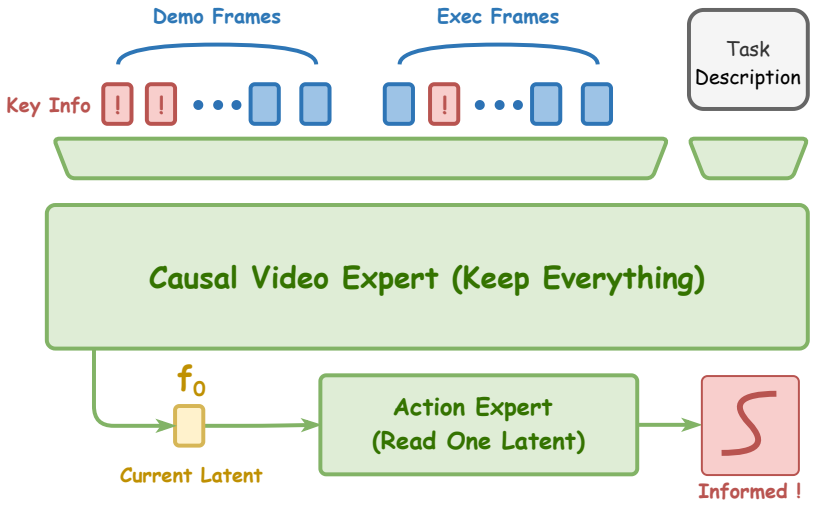


Figure 1: **Keep everything, read one latent.** The causal video expert keeps every latent of the episode and integrates this history into the current latent f_0.

3.1 PROBLEM SETUP

A world-action model encodes observations into latents with the encoder of a pretrained video generator and couples two experts that share attention: a video expert F_θ and an action expert A_ϕ. Following Fast-WAM (Yuan et al., 2026), we use future-video rendering as a proxy task during training, implemented as future-latent prediction. At inference, we generate actions without rendering future video. Training fits the factorized prediction

$$p_\theta\big(\mathbf{z}_{t+1:t+K} \mid \ell,\ \mathcal{O}_{\le t}\big)\ \pi_{\theta,\phi}\big(\mathbf{a}_t \mid \ell,\ \mathcal{O}_{\le t},\ q_t\big), \tag{1}$$

and deployment samples only the second factor, the policy

$$\mathbf{a}_t \sim \pi_{\theta,\phi}\big(\cdot \mid \ell,\ \mathcal{O}_{\le t},\ q_t\big). \tag{2}$$

Here ℓ is the language instruction, q_t the proprioceptive state, $\mathbf{a}_t$ the action chunk executed until the next control step, $\mathbf{z}_{t+1:t+K}$ the latents of the next K observations, and $\mathcal{O}_{\le t}$ all observations up to control step t.

3.2 VERBATIM EPISODE MEMORY

The video generator's causal VAE maps the sampled frames of each re-planning interval into one latent, and the stream grows by one latent per control step. We index latents by their position in the stream, so that at control step t the context is

$$\mathbf{z}_{\le t} = (z_1, \dots, z_{t-1}, z_t), \qquad f_0 \equiv z_t, \tag{3}$$

where f_0 is the *current latent*, which holds the newest observations, and $\mathbf{z}_{<t}$ is the history. Here, *verbatim* is defined at the latent level: every latent produced by temporal sampling and native VAE encoding (§3.6) is retained, clean and at its native latent resolution.

The demonstration is treated as part of the same stream. Demonstration and execution latents therefore share one representation, and the causal mask treats the boundary between them like any other. A demonstrated motion is stored at the same temporal resolution as the robot's own execution, one latent per re-planning interval.

3.3 Causal History Integration in the Video Expert

The video expert processes the stream under a block-causal mask: every token of z_i attends to all tokens of $\mathbf{z}_{\leq i}$. The representation of the current latent,

$$\mathbf{h}_t = F_\theta\big(f_0 \mid \mathbf{z}_{<t},\ \ell\big), \tag{4}$$

is therefore a function of the entire episode, while its size stays fixed at 98 tokens. History reaches $\mathbf{h}_t$ only through the backbone's own self-attention. We call $\mathbf{h}_t$ a *history-conditioned predictive state*: a fixed-size representation that is computed from the whole episode and optimized to support prediction of future observations and expert actions (§3.5).

3.4 A Fixed-Size Interface for Action Decoding

The inputs of the action expert are the noised action chunk and the proprioceptive state q_t; all visual information reaches it through shared attention. Video tokens attend block-causally to video tokens (§3.3) and never to action tokens. Action tokens attend to one another and, among visual tokens, only to the 98 tokens of f_0. Every visual influence of the history on the action must therefore pass through $\mathbf{h}_t$.

The mask factorizes the policy of Eq. (2) into two stages,

$$\mathbf{z}_{\leq t} \xrightarrow{F_\theta} \mathbf{h}_t \xrightarrow{A_\phi} \mathbf{a}_t, \qquad \mathbf{a}_t \sim A_\phi\big(\cdot \mid \mathbf{h}_t,\ q_t\big). \tag{5}$$

The video expert folds non-Markovian episode history into the predictive state $\mathbf{h}_t$. The action expert decodes actions through a Markovian interface: its output depends on the history only through $\mathbf{h}_t$.

3.5 Training Objective

VEM adds no memory-specific loss or label. Each training example is a control step t of an expert episode, with the context built exactly as at deployment: the clean prefix $\mathbf{z}_{\leq t}$, demonstration included, encoded in one forward pass. The next K latents of the episode, $\mathbf{y} = \mathbf{z}_{t+1:t+K}$, are noised, appended after f_0, and denoised by the video expert with the generative objective of its pretraining. These future latents attend to the clean prefix, f_0 included, in every layer, whereas no prefix token attends to them. Both $\mathbf{y}$ and the expert action chunk $\mathbf{a}_t$ are trained with flow matching (Lipman et al., 2023). For a target $\mathbf{x}$, noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and flow time $\tau \in [0, 1]$, the noised input is $\tilde{\mathbf{x}} = \tau\mathbf{x} + (1-\tau)\boldsymbol{\epsilon}$, and the network regresses the velocity $\mathbf{x} - \boldsymbol{\epsilon}$:

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \lambda\,\mathcal{L}_{\text{vid}}, \qquad \begin{aligned} \mathcal{L}_{\text{act}} &= \mathbb{E}\left\|u_\phi(\tilde{\mathbf{a}}_t, \tau' \mid \mathbf{h}_t, q_t) - (\mathbf{a}_t - \boldsymbol{\epsilon}')\right\|^2, \\ \mathcal{L}_{\text{vid}} &= \mathbb{E}\left\|v_\theta(\tilde{\mathbf{y}}, \tau \mid \mathbf{z}_{\leq t}, \ell) - (\mathbf{y} - \boldsymbol{\epsilon})\right\|^2. \end{aligned} \tag{6}$$

Only the future latents and the action chunk are noised; the prefix is clean in both terms. Both terms shape $\mathbf{h}_t$. The video loss supplies a predictive training signal through the attention of the future latents, which also read earlier prefix latents directly. The action loss reaches the video expert only through $\mathbf{h}_t$, because the action expert reads nothing else, and so trains $\mathbf{h}_t$ to carry the history the action needs. Supervision consists of expert actions and the episode's future video targets; the task instruction is a conditioning input. There are no keyframe labels, subgoal annotations, or memory targets.

3.6 Implementation

The video expert is Wan2.2-TI2V-5B (Wan Team et al., 2025) with its causal VAE, and the action expert is a 1024-dimensional flow-matching transformer, 1B parameters. The model receives two 224×448 views. Frames are sampled every four environment steps, and the causal VAE maps every four sampled frames into one latent, so each latent spans 16 environment steps. Figure 2 reports end-to-end control-step latency on one H200, including new-latent encoding and action denoising, for histories of up to 300 video latents (4,800 environment steps).

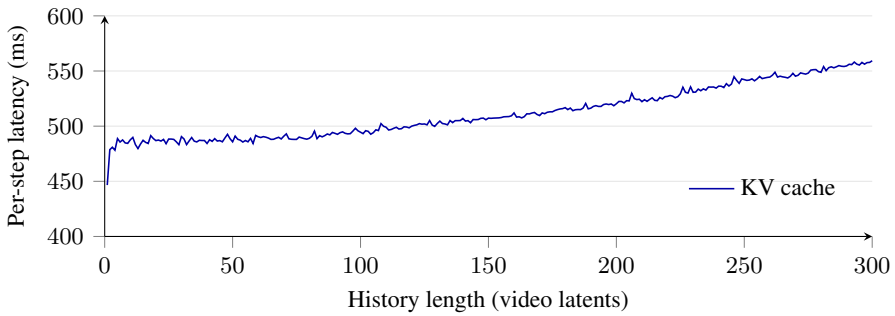


Figure 2: **KV-cache latency over 4,800 environment steps.** End-to-end control-step latency on one H200 GPU, with one video latent per 16 environment steps.

4 EXPERIMENTS

We ask four questions. **(Q1)** What is gained by retaining the full causal episode? **(Q2)** Can the model retrieve action-relevant information from deep episode history? **(Q3)** What does visual episode memory preserve beyond oracle grounded subgoals? **(Q4)** Can history integration be separated from action decoding?

4.1 EXPERIMENTAL SETUP

Benchmark. RoboMME (Dai et al., 2026) contains 16 tabletop tasks in four suites: *Counting* (BinFill, PickXtimes, SwingXtimes, StopCube), *Permanence* (VideoUnmask, ButtonUnmask, and their Swap variants), *Reference* (PickHighlight, VideoRepick, VideoPlaceButton, VideoPlaceOrder), and *Imitation* (MoveCube, InsertPeg, PatternLock, RouteStick). Nine tasks begin with a demonstration video. The Imitation instructions contain no task variables, so the intended behavior can be identified only from the demonstration. The training set contains 100 expert episodes per task (1,600 episodes, 769k frames).

Protocol. Each model is a single checkpoint evaluated on all 16 tasks with the 50 fixed test-split environment seeds per task and `max_steps` = 1300. For OURS, we report mean $\pm$ std over three inference random seeds (7, 42, 0) applied to the same trained checkpoint, not three independent training runs. An inference seed initializes the PyTorch, NumPy, and Python random number generators once at evaluation start and mainly changes the initial noise of action denoising. Environment seeds are fixed across inference seeds, so the standard deviation measures inference-time policy stochasticity rather than training variance.

Variants. All variants use the VEM WAM backbone (§3.6) and differ in their context:

- f_0**-only**: the current latent without visual history.
- **Budget**: k_d=k_e=6 sampled demonstration and execution latents together with f_0 (13 latents, 1,274 tokens). After VAE encoding, six latents are sampled at evenly spaced positions across the full demonstration latent sequence. The execution context consists of the six most recent execution latents preceding f_0; f_0 itself is included separately. It shares the training recipe and f_0-only action read of OURS and differs only in how much history is retained.
- **OURS**: the full causal visual episode (§3.2).
- **Oracle-SG (**f_0**-only)**: f_0-only, trained and evaluated with ground-truth grounded subgoal text whose referents are resolved to pixel coordinates and which is updated online from simulator state.

External baselines. We compare with $\pi_{0.5}$ (Physical Intelligence et al., 2025), FRAMESAMP-Modul, FRAMESAMP-Expert, and MemER (Sridhar et al., 2026), using the results reported by RoboMME (Dai et al., 2026). For Q3 we also report GROUNDSG-Oracle, which supplies ground-truth grounded subgoals to the $\pi_{0.5}$ VLA backbone.

Table 1: **RoboMME success rate (%), suite averages.** OURS reports mean ± std over three inference seeds. Budget is a fixed-budget causal context with the same f_0-only action interface as OURS; f_0-only is a no-history lower bound.

Method	Counting	Permanence	Reference	Imitation	**Avg.**
$\pi_{0.5}$	28.8	17.0	17.2	8.8	17.9
FRAMESAMP-Expert	**66.8**	25.2	24.1	29.0	36.3
MemER	48.8	53.2	38.0	29.5	42.4
FRAMESAMP-Modul	65.2	25.1	36.3	51.4	44.5
VEM WAM f_0-only	17.5	13.0	13.0	9.0	13.1
VEM WAM Budget	43.0	43.0	49.0	78.0	53.3
OURS	52.5±1.3	**56.8**±0.8	**62.7**±0.8	**78.2**±2.9	**62.5**±1.2

Table 2: **RoboNIAH behavioral retrieval across temporal depth.** Each entry reports success / wrong-target / timeout counts over 16 matched scenes.

Inserted delay Δ (steps)	0	64	256	512	1024	2048	4096
Needle distance (latents)	0	4	17	33	65	129	257
Needle	16/0/0	16/0/0	16/0/0	15/1/0	15/1/0	11/3/2	1/0/15
No needle	6/10/0	7/9/0	5/11/0	7/9/0	7/9/0	6/10/0	0/0/16

4.2 Q1: WHAT IS GAINED BY RETAINING THE FULL CAUSAL EPISODE?

We first ask whether removing the fixed history budget improves performance. The same-backbone variants range from no history, through a bounded causal context, to the complete sampled episode (Table 1).

Retaining the complete episode adds 9.2 percentage points over Budget on the same backbone, with the largest gains on Permanence and Reference. Imitation changes little, suggesting that densely informative demonstrations can survive fixed-budget sampling, whereas sparse event retrieval benefits more from complete history.

OURS exceeds FRAMESAMP-Modul on 11 of 16 tasks (Table 6), with large gains on tasks requiring earlier visual evidence.

4.3 Q2: LONG-RANGE RETRIEVAL

Aggregate gains show that history helps, but not how deep a specific past observation remains usable. We therefore construct **RoboNIAH**, a controlled behavioral retrieval test based on ButtonUnmask.

Protocol. At reset, the object is visible inside one of three containers. This initial observation is the *needle*: direct visual evidence of which container will later hold the hidden object. A scripted controller presses the button, which closes all containers, and then hovers above the button for Δ environment steps, adding task-irrelevant history without revealing the target; control then passes to the learned policy. During the scripted prefix, observations enter the model through the same deployment path as in normal control, one latent per 16 steps, and the predicted actions are discarded, so the intervening history consists of real simulator observations rather than synthetic latents. We evaluate $\Delta \in \{0, 64, 256, 512, 1024, 2048, 4096\}$ on the same 16 scenes. In a matched *no-needle* control, the model does not receive the initial observation revealing the object's location, while the button press, hover, and policy phase are unchanged. Each rollout ends in *success*, *wrong target*, or *timeout*.

The needle advantage persists through 65 latents and remains visible at 129 latents (Table 2, Figure 3), long after the target stopped being visible. This is more than twice as deep as the longest ButtonUnmask training context, and the advantage at 129 latents lies beyond the longest episode in the multitask dataset.

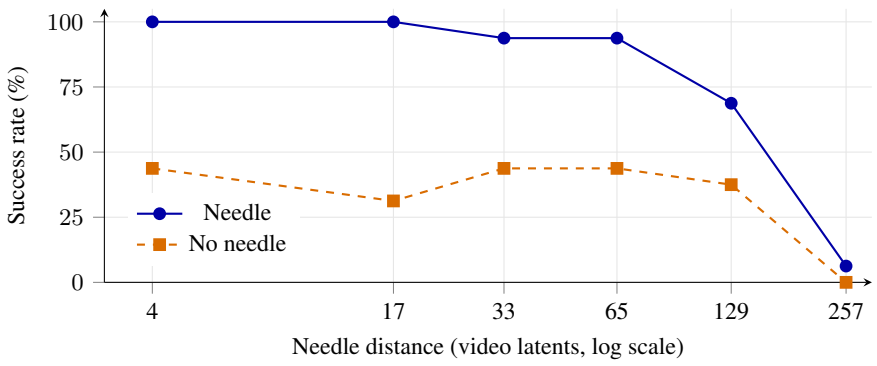


Figure 3: **RoboNIAH retrieval across temporal depth.** Success rates of ButtonUnmask over 16 matched scenes at each distance.

Table 3: **Imitation suite: oracle grounded subgoals vs. verbatim visual episode memory** (success %). The VEM WAM rows share backbone and training recipe and differ in their visual-history and subgoal inputs. OURS averages three inference seeds of one checkpoint. The $\pi_{0.5}$ row provides a comparison on a language-conditioned backbone. Best result per column in **bold**.

Context / memory	Backbone	MoveCube	InsertPeg	PatternLock	RouteStick	**Avg.**
GT grounded subgoal	$\pi_{0.5}$ VLA	**87.8**	15.6	**97.0**	55.6	64.0
none (f_0-only)	VEM WAM	20.0	0.0	10.0	6.0	9.0
GT grounded subgoal (f_0-only)	VEM WAM	80.0	20.0	60.0	38.0	49.5
verbatim visual episode (OURS)	VEM WAM	73.3	**52.0**	88.0	**99.3**	**78.2**

4.4 Q3: VISUAL HISTORY VERSUS ORACLE SUBGOALS

In the Imitation suite the demonstration carries the task-critical information, since the instructions contain no task variables. We ask how much of this information survives when the visual episode is replaced by a compact, semantically correct oracle abstraction. Oracle-SG (f_0) shares the backbone and training recipe of OURS but receives no visual history, only the ground-truth grounded subgoal (§4.1). To assess whether the gap reflects limited language grounding in our backbone, we also compare against GROUNDSG-Oracle (Dai et al., 2026), which supplies the same form of subgoal to the language-conditioned $\pi_{0.5}$ VLA backbone.

Oracle subgoals underperform visual history on average on both backbones (Table 3). The $\pi_{0.5}$ oracle exceeds OURS on MoveCube and PatternLock, whereas both oracles trail on InsertPeg and RouteStick. The latter tasks depend on insertion geometry and route shape, making missing motion information a plausible explanation.

The tested grounded subgoal specifies a target object and location, whereas a visual demonstration can also convey trajectory shape, approach geometry, ordering, and contact. The results suggest that the existing grounded-subgoal representation does not replace the action-relevant information available in visual demonstrations. Differences in grounding, optimization, and backbone capability remain possible contributors to the observed gaps.

4.5 Q4: CAN HISTORY INTEGRATION BE SEPARATED FROM ACTION DECODING?

The mask of §3.4 lets the action expert read only the 98 tokens of f_0, so every visual influence of the history must pass through the predictive state $\mathbf{h}_t$. We test whether this narrow read costs performance on the 13-latent Budget context. The two variants share the training recipe and visual context and differ only in action-to-visual visibility: in the *full-prefix* variant, action tokens attend to all 1,274 visual-context tokens; in the f_0*-read* variant, only to the 98 tokens of f_0, which is the Budget model of Table 1. Video-to-video and action-to-action attention are unchanged.

Table 4: **Direct history access for action decoding** (success %). Both variants use the same 13-latent Budget visual context and training recipe; the f_0-only row is the Budget model of Table 1.

Action reads	Count.	Perm.	Ref.	Imit.	**Avg.**
full prefix (1,274 tok.)	41.2	39.5	45.0	**79.5**	51.3
f_0 only (98 tok.)	**43.0**	**43.0**	**49.0**	78.0	**53.3**

The paired task-level comparison detects no statistically significant difference between the two reads (Table 4; f_0-only minus full prefix, 16 tasks: t=1.26, p≈0.23, 95% CI $[-1.3, +5.1]$ percentage points). This comparison does not establish equivalence between the two reads.

4.6 Limitations of the Current Model

The two Swap variants, StopCube, and BinFill remain below 30% success. Their decisive variable must be continually *updated* over time: a running count, a stopping condition, or a location changed by a swap. By contrast, many of the largest gains from full history occur when the policy must recover information from a specific earlier observation, the behavior that RoboNIAH isolates. The current model therefore shows a clear empirical asymmetry between *episodic retrieval* and *persistent state tracking*. RoboNIAH also exposes a second failure mode: at 257 latents, closed-loop execution collapses before retrieval can be measured. Understanding the respective scaling limits of retrieval, predictive-state formation, and closed-loop control remains an open direction.

Oracle grounded-subgoal policies perform substantially better on BinFill and the Swap tasks (Table 6). A concrete next step is to combine visual episode memory with explicit state tracking for running counts and changing object locations.

5 Conclusion

VEM makes the full visual episode a working memory for world-action models. Demonstration and execution share one causal stream that retains every sampled video latent. Trained on future video and expert actions, the video backbone integrates this history into the current latent, which serves as the action expert's fixed-size visual interface. On RoboMME, VEM reaches 62.5% success, gaining 9.2 percentage points over a fixed-budget context on the same backbone. A controlled retrieval test shows that an early observation can guide behavior beyond the longest training episode. The same stream supports 78.2% success on imitation tasks, surpassing both grounded-subgoal oracle baselines and demonstrating the value of visual demonstrations as a direct source of action guidance. These results establish a practical design principle for WAM memory: separate how much the policy retains from how much the action expert reads. Keep the episode, integrate history in the video backbone, and decode actions from the current latent.

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.

Reproducibility statement

Sections 3 and 4.1 describe the architecture, objectives, dataset, and evaluation protocol, and Appendix A lists the training and inference hyperparameters of Ours. Reported variation for Ours is across inference seeds of one checkpoint, not independent training runs.

REFERENCES

Jingjing Chen, Hongjie Fang, Chenxi Wang, Shiquan Wang, and Cewu Lu. History-aware visuomotor policy learning via point tracking. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Egor Cherepanov, Nikita Kachaev, Alexey K. Kovalev, and Aleksandr I. Panov. Memory, benchmark & robots: A benchmark for solving complex tasks with reinforcement learning. In *Proceedings of the 7th Robot Learning Workshop at ICLR*, 2025.

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

Yan Duan, Marcin Andrychowicz, Bradly Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in Neural Information Processing Systems*, 2017.

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 *International Conference on Machine Learning*, 2025.

Gemini Robotics Team, Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, et al. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. *arXiv preprint arXiv:2510.03342*, 2025.

Huiwon Jang, Sihyun Yu, Heeseung Kwon, Hojin Jeon, Younggyo Seo, and Jinwoo Shin. ContextVLA: Vision-language-action model with amortized multi-frame context. *arXiv preprint arXiv:2510.04246*, 2025.

Yunfan Jiang, Yevgen Chebotar, Ruijie Zheng, Fengyuan Hu, Yunhao Ge, Jimmy Wu, Tianyuan Dai, Scott Reed, Li Fei-Fei, Yuke Zhu, and Linxi Fan. RoboTTT: Context scaling for robot policies. *arXiv preprint arXiv:2607.15275*, 2026.

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

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.

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

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

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025.

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

Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Pengju An, Xiaoqi Li, Kaichen Zhou, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. RoboMamba: Efficient vision-language-action model for robotic reasoning and manipulation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 40085–40110, 2024.

Liheng Ma, Rui Heng Yang, Zhanguang Zhang, Mateo Clemente, Ziwen Hu, Tongtong Cao, and Yingxue Zhang. Faster-WAM: Do world action models need deep action modules? *arXiv preprint arXiv:2608.02365*, 2026.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning*, 2021.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Hongyu Qu, Jianzhe Gao, Xiaobin Hu, Shaohuan Yang, Xinlei Yu, Rui Yan, Wenguan Wang, Xiangbo Shu, and Shuicheng Yan. Dual latent memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2607.07608*, 2026.

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*, 2026.

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.

Jun Sun, Boyu Yang, Jiahao Zhang, Ning Ma, Chencheng Wu, Siqing Zhang, Yiou Huang, Qiufeng Wang, Shan Liang, and Yaran Chen. TempoFit: Plug-and-play layer-wise temporal KV memory for long-horizon vision-language-action manipulation. *arXiv preprint arXiv:2603.07647*, 2026.

Marcel Torne, Andy Tang, Yuejiang Liu, and Chelsea Finn. Learning long-context diffusion policies via past-token prediction. *arXiv preprint arXiv:2505.09561*, 2025.

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

Wan Team, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

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

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*, 2025.

A TRAINING HYPERPARAMETERS

Table 5 lists the architecture, training, and inference settings of OURS.

Table 5: **Hyperparameters of OURS.**

Setting	Value
Video expert	Wan2.2-TI2V-5B video DiT (30 layers, width 3072, 24 heads)
Action expert	DiT with mixture-of-transformers joint attention to the video DiT (depth 30, width 1024, 24 heads $\times$ 128)
Observations	Two cameras (third-person and wrist), 224×224 each, concatenated to 224×448
Proprioception	One token in the cross-attention context
Context	Full causal stream of all past latents (98 tokens per latent); the action expert reads only the current latent
Action space	Absolute joint angles (7 joints and gripper)
Action horizon	32 steps, re-planned every 16
Flow matching	Timestep shift 5; 10 denoising steps at inference
Optimizer	AdamW (β_1=0.9, β_2=0.95), weight decay 0.01
Learning rate	1×10^{-4}, 5% warmup, cosine decay to 1×10^{-6}
Gradient clipping	Norm 1.0
Batch size	$\leq$16 per GPU (token-budget bucketing, 98,304 tokens per GPU), $\leq$128 global
Training steps	76,260 (20 epochs)
Precision	bf16
Parallelism	DeepSpeed ZeRO-1 on 8 H200 GPUs

B PER-TASK RESULTS

Table 6 compares OURS with prior methods on every task.

Table 6: **Per-task success rate (%) of prior methods and OURS.** All rows except OURS are taken from Dai et al. (2026). The oracle row receives ground-truth subgoals. The best result among non-oracle end-to-end policies is shown in **bold**.

	Counting				Permanence			
Method	Bin Fill	Pick Xtimes	Swing Xtimes	Stop Cube	Video Unmask	Button Unmask	Video Unmask Swap	Button Unmask Swap
GROUNDSG-Oracle	85.8	100.0	100.0	49.7	98.8	95.0	99.2	80.2
FRAMESAMP-Modul	39.6	**87.3**	92.0	**42.0**	32.7	25.1	24.4	18.2
FRAMESAMP-Expert	**57.3**	86.2	**94.7**	28.9	31.8	25.8	22.9	20.2
$\pi_{0.5}$	30.0	42.9	35.6	6.7	20.4	22.2	18.7	6.7
MemER	56.7	79.3	59.3	0.0	81.3	72.0	**38.0**	21.3
OURS	19.3	72.0	92.7	26.0	**92.7**	**84.7**	23.3	**26.7**

	Reference				Imitation				
Method	Pick Highlight	Video Repick	Video PlaceButton	Video PlaceOrder	Move Cube	Insert Peg	Pattern Lock	Route Stick	**Avg.**
GROUNDSG-Oracle	83.3	97.3	100.0	100.0	87.8	15.6	97.0	55.6	84.1
FRAMESAMP-Modul	22.9	30.4	60.0	32.0	77.8	7.6	53.6	66.7	44.5
FRAMESAMP-Expert	19.1	23.1	30.0	24.2	**83.1**	2.0	13.6	17.1	36.3
$\pi_{0.5}$	11.3	0.4	31.1	25.8	26.0	1.6	2.9	4.7	17.9
MemER	**70.7**	25.3	30.0	26.0	82.7	6.7	16.7	12.0	42.4
OURS	68.0	**42.0**	**78.7**	**62.0**	73.3	**52.0**	**88.0**	**99.3**	**62.5**