SHARP-WAM: SPARSE HIERARCHICAL ATTENTION WITH PROBE TOKENS FOR BOUNDED-CONTEXT WORLD ACTION MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) present a compelling framework for robotic manipulation via video prediction. However, scaling WAMs to long-horizon tasks is hindered by the dense attention complexity of foundation backbones. Existing approaches either compromise non-Markovian dependencies via truncated sliding windows, or introduce external memory plug-ins that risk complicating the pretrained backbone. Leveraging the insight that long-horizon action modeling relies on sparse historical interactions, we propose SHARP-WAM, a **S**parse **H**ierarchical **A**ttention framework with Lea**r**nable **P**robe Tokens for bounded-context World Action Models. This framework introduces a compact set of learnable probe tokens serving as contextualized descriptors and routers. Historical key-value states are retained within a native, fixed-capacity KV Set via online utility-driven eviction, thereby enabling horizon-invariant attention inference cost. Building upon this bounded backbone-native memory, we devise a hierarchical sparse attention scheme. Temporal Top-K routing is first performed via probe tokens to select relevant candidates from the Fixed KV Set, followed by camera-aware spatial partitioning. This dual-level sparsity substantially curbs long-horizon computational overhead while preventing cross-view feature corruption and preserving fine-grained semantics. Extensive experiments demonstrate that SHARP-WAM outperforms existing methods on long-horizon tasks, achieving success rates of 76.0% on RMBench, 47.0% on LIBERO-Mem, and 75.0% in real-world trials, while remaining competitive on general-purpose benchmarks. Project page.

1 INTRODUCTION

Vision-language-action models (VLAs) have served as a core foundation for building generalist robot policies, leveraging scalable vision-language backbones to map language instructions and visual observations to executable action chunks (Black et al., 2025a; Physical Intelligence et al., 2026; Kim et al., 2025; NVIDIA et al., 2025). However, standard VLAs remain weakly grounded in fine-grained physical dynamics, primarily relying on constrained action-centric supervision. World Action Models (WAMs) offer a promising route to bridge this gap by jointly learning executable action chunks and future world states. Existing WAMs span generative video prediction (Yuan et al., 2026; Ye et al., 2026; Kim et al., 2026; MotuBrain Team et al., 2026) and latent dynamics modeling (Luo et al., 2026). Nevertheless, existing WAMs struggle in long-horizon tasks, where dense history contexts impose rapidly growing computation and memory costs on large foundation backbones.

Under the quadratic cost of dense attention in Diffusion Transformers (DiT) (Peebles & Xie, 2023), practical WAM deployments often bound their active histories using sliding-window caching to sustain continuous rollouts (Li et al., 2026a; Cai et al., 2026). This trade-off improves tractability, while inherently weakening long-horizon temporal awareness. Several memory-augmented alternatives have therefore been explored, each with distinct limitations. External memory banks retain selected historical evidence (Shi et al., 2026b;a; Sun et al., 2026), but require dedicated retrieval modules as external plug-ins that complicate the pretrained architecture. Other approaches introduce auxiliary supervision to learn heuristic abstractions, such as sub-task language decomposition (Torne et al., 2026; Yang et al., 2026b) or keyframe utilization (Zeng et al., 2026; Yang et al., 2026a), and most of

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.

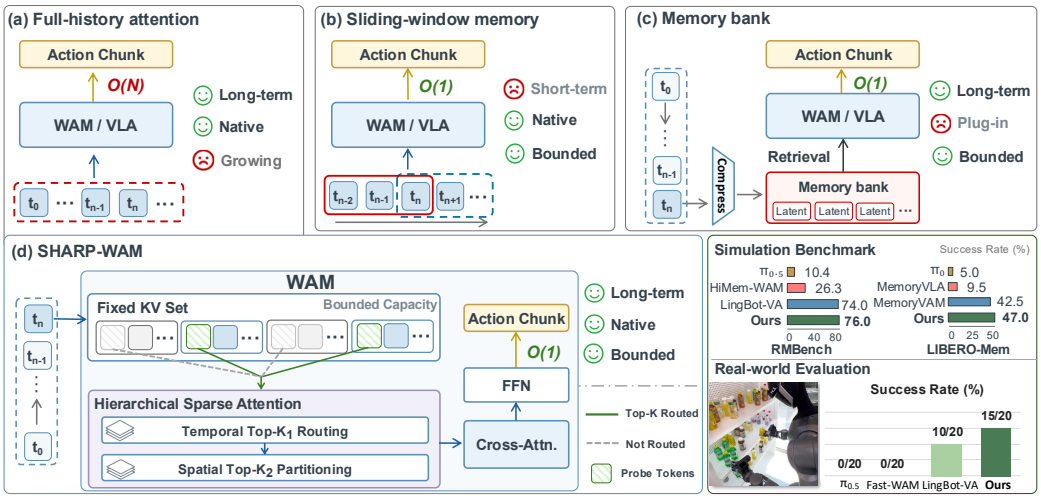


Figure 1: **Overview.** Existing WAMs face an inherent dilemma between maintaining long-term memory and managing horizon-scaling inference complexity. While memory-bank retrieval alleviates this trade-off, it relies on dedicated retrieval modules added as external plug-ins to the backbone. In contrast, SHARP-WAM introduces a long-horizon world action model that achieves native in-backbone memory retention alongside horizon-invariant per-step $O(1)$ inference complexity, delivering competitive performance across diverse tasks.

these methods rely heavily on additional annotations and complex designs. Alternatively, compressing history embeddings into compact latent representations reduces computational overhead (Shah et al., 2026; Jiang et al., 2026), yet introduces an information bottleneck that can discard fine-grained evidence.

Sparse attention has become a leading end-to-end approach to native long-context modeling in large language models (Lu et al., 2025; Yuan et al., 2025) and video diffusion models (Xu et al., 2026; Wu et al., 2026; Zhang et al., 2025). Crucially, the success of aforementioned memory retrieval and keyframe selection suggests that long-horizon manipulation is inherently sparse, where actions typically depend on selective interactions with the past rather than consuming the entire history. However, exploiting this long-horizon spatiotemporal sparsity natively within WAM backbones remains underexplored. As illustrated in Figure 1, we propose SHARP-WAM, a **S**parse **H**ierarchical **A**ttention framework with Lea**r**nable **P**robe Tokens for World Action Modeling, which unifies backbone-native hierarchical sparse attention with bounded-context $O(1)$ inference complexity.

SHARP-WAM mitigates the aforementioned dilemma through three complementary innovations. First, we introduce probe tokens, a set of learnable queries designed to compactly summarize multi-camera observations while preserving view-specific semantics. Unlike mean pooling that conflates distinct views, probe tokens distill informative context into compact descriptors that further serve as routing interfaces for sparse retrieval. Second, a fixed-capacity KV Set is proposed to retain historical KV states. Constrained by causal attention masks, incoming probe tokens attend to preceding ones to recursively accumulate temporal context. Meanwhile, an online eviction policy identifies redundant pairs via probe similarity then discards obsolete states based on an Exponential Moving Average (EMA) of historical retrieval utility, maintaining an $O(1)$ cache size. Finally, we establish a coarse-to-fine hierarchical sparse attention mechanism. Probe tokens are utilized as a query-relevant routing interface to perform layer-wise Top-K sparse retention over the Fixed KV Set. The selected historical KV states are subsequently processed through camera-aware partitioning, enforcing structured spatiotemporal attention to prevent cross-view feature corruption.

We conduct extensive evaluations across RMBench, LIBERO-Mem, LIBERO, and real-world manipulation, demonstrating superior long-horizon performance and strong general competitiveness.

In summary, our contributions are threefold:

- We propose SHARP-WAM, pioneering backbone-native sparse attention in World Action Models to enable end-to-end training and horizon-invariant inference complexity for long-horizon manipulation.

- We design a hierarchical sparse architecture that couples learnable probe tokens with a Fixed KV Set, bridging temporal Top-K routing and camera-aware spatial sparsity.
- SHARP-WAM outperforms the compared baselines on RMBench, LIBERO-Mem, and the real-world task, while remaining competitive on LIBERO.

2 Related work

2.1 World action models

Vision-language-action (VLA) models map visual observations and language instructions to robot actions, whereas world action models (WAMs) also predict future states to learn action-conditioned dynamics, often building on pretrained video models (Li et al., 2026a; Kim et al., 2026; MotuBrain Team et al., 2026; Luo et al., 2026). DreamZero jointly predicts visual states and actions with a video diffusion backbone (Ye et al., 2026); Fast-WAM instead uses video prediction only during training and omits generation at inference (Yuan et al., 2026). Both focus on how future modeling improves action prediction, leaving efficient storage and sparse routing of growing history less explored.

2.2 Memory-centric vision-language-action and world-action models

MemoryVLA and MemoryVLA++ store task-relevant history in dedicated memory banks (Shi et al., 2026b;a). HiMem-WAM uses a boundary-aware gate to write compact task states into a memory bank at skill transitions (Sun et al., 2026). These methods rely on dedicated banks, compression, or specialized updates rather than sparse access to native, high-fidelity historical key–value (KV) states.

2.3 Sparse attention in large language and video generation models

Sparse attention scales long-context models by selecting only relevant history. Native Sparse Attention combines compressed, selected, and local branches, whereas MoBA routes queries to a few KV blocks (Yuan et al., 2025; Lu et al., 2025). Video DiTs exploit their $T \times H_{\text{img}} \times W_{\text{img}}$ structure through blockwise, coarse-to-fine routing (Xi et al., 2025; Zhang et al., 2025; Wu et al., 2026). However, using sparse attention as an end-to-end memory mechanism over native WAM KV caches remains underexplored, while direct transfer risks mixing head- and wrist-camera views that do not share a spatial grid.

3 SHARP-WAM

3.1 Problem Formulation and Overview

Problem Formulation. We formulate long-horizon robotic manipulation as a sequential decision-making process under bounded computational constraints. At each control step t, a world action model π_θ receives the observation history $\mathcal{O}_{\le t} = \{o_0, \dots, o_t\}$ and a language instruction y. During training, it jointly predicts an executable action chunk $\mathcal{A}_t$ and a future visual sequence $\mathcal{V}_t$:

$$(\mathcal{A}_t, \mathcal{V}_t) = \pi_\theta(\mathcal{O}_{\le t}, y)\,, \tag{1}$$

where $\mathcal{A}_t = \{a_t, \dots, a_{t+h_a-1}\}$ spans an action horizon h_a, and $\mathcal{V}_t = \{v_{t+1}, \dots, v_{t+h_v}\}$ spans a visual horizon h_v. At inference, only the action prediction is executed, while explicit visual generation is omitted. Although the auxiliary visual prediction grounds physical dynamics, dense full-history self-attention is quadratic in sequence length, while KV-cached incremental attention remains linear per decision. SHARP-WAM instead uses sparse historical interactions to capture long-range dependencies.

Model Overview. SHARP-WAM builds on a dual Mixture-of-Transformers (MoT) architecture designed for joint visual and physical reasoning. The architecture comprises a video-expert DiT dedicated to modeling visual dynamics and a lightweight action-expert DiT tailored to action execution. Their token streams first interact through joint self-attention, where SHARP-WAM applies hierarchical sparsity, and then undergo cross-attention. Language embeddings encoded by a text encoder condition both DiT backbones. At test time, the video backbone provides physically grounded

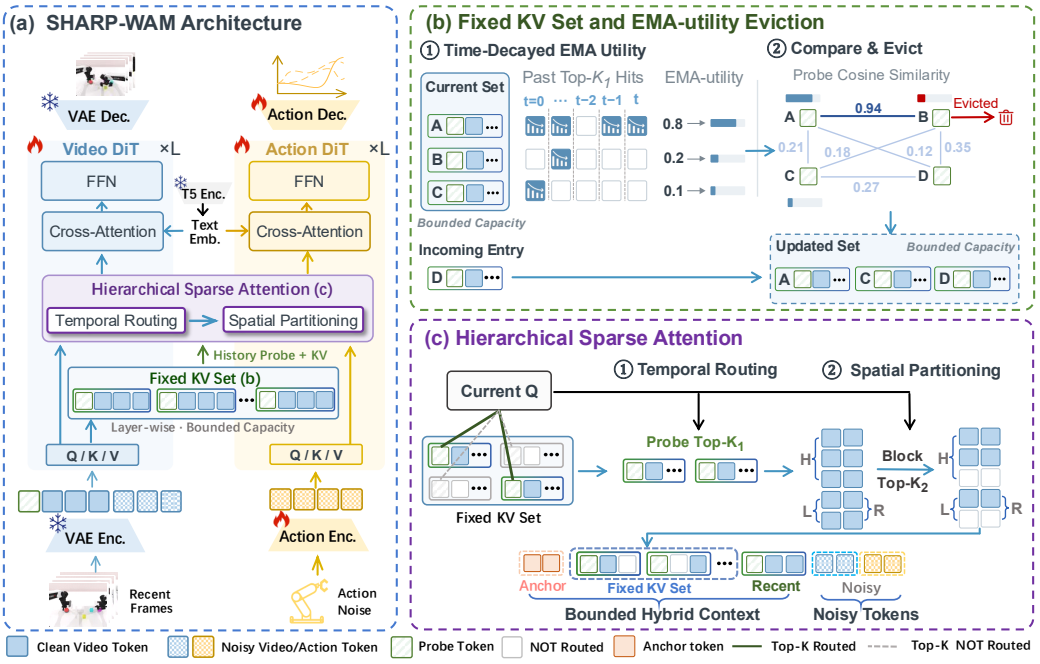


Figure 2: **Overview of SHARP-WAM.** (a) SHARP-WAM couples video and action DiTs through a layer-wise, fixed-capacity KV set and hierarchical sparse attention, where learnable probe tokens serve as compact descriptors and routing interfaces. (b) Historical KV states are maintained within a Fixed KV Set via exponential-moving-average (EMA) utility eviction, ensuring horizon-invariant memory complexity. (c) Hierarchical sparse attention first performs temporal Top-K_1 routing over probe tokens and then camera-aware spatial Top-K_2 partitioning, producing a bounded hybrid context for attention.

feature representations while bypassing explicit pixel synthesis, allowing the action expert to execute with low control latency. To support continuous long-horizon rollouts, historical KV states are dynamically maintained within a fixed-capacity KV set. This bounded set provides the memory foundation for the hierarchical sparse attention mechanism developed below.

3.2 Learnable Probe Tokens as Contextualized Descriptors and Routers

Long-horizon action modeling requires lightweight history descriptors for fixed-capacity KV maintenance and sparse routing. Directly scoring all historical visual keys undermines the efficiency of sparse retrieval, whereas the parameter-free mean pooling used by prior sparse-attention methods (Lu et al., 2025; Xu et al., 2026; Wu et al., 2026) may mix camera-specific semantics and dilute local manipulation cues. Inspired by learnable query and summary-token mechanisms (Carion et al., 2020; Jaegle et al., 2021; Mao et al., 2026), we introduce probe tokens that adaptively summarize each observation and serve as routing interfaces to its original visual KV states.

Let $\mathbf{X}_t^0$ denote the clean visual tokens obtained by encoding and patchifying the multi-view observation at frame t, at the input to Transformer layer $\ell = 0$. We append a small set of learnable probe tokens to each frame:

$$\begin{aligned} &\mathbf{X}_t^0 \in \mathbb{R}^{N\times d}, \\ &\mathbf{P}_t^0 = \mathbf{P}_{\text{init}} \in \mathbb{R}^{N_p\times d}, \quad N_p \ll N. \end{aligned} \tag{2}$$

where N and N_p are the numbers of visual and probe tokens per frame, respectively, and d is the video backbone's hidden dimension. The shared initialization $\mathbf{P}_{\text{init}}$ is updated end-to-end by the video DiT's native Transformer blocks, producing observation- and history-dependent probes without an auxiliary encoder.

As shown in Figure 3, probes at frame t attend to the initial visual anchor tokens, retained earlier probes, and current visual and probe tokens. Action tokens attend to current probes, but not vice versa.

The causal mask blocks direct access to non-anchor historical visual tokens. Let $\mathcal{C}^{\ell}_{t,P}$ denote the allowed KV context at layer ℓ. The probe update is

$$\mathbf{P}^{\ell+1}_{t} = \mathcal{B}^{\ell}_{v}\big(\mathbf{P}^{\ell}_{t}; \mathcal{C}^{\ell}_{t,P}\big)\,, \tag{3}$$

where $\mathcal{B}^{\ell}_{v}$ is the video backbone's layer-ℓ Transformer block. For a backbone with L blocks, $\ell \in \{0, \ldots, L-1\}$ and $\mathbf{P}^{L}_{t}$ denotes the final-layer probes. By restricting direct attention to non-anchor visual history, the causal mask channels historical context through compact probes, whose layer-wise updates yield contextualized descriptors. The corresponding visual KV states are retained alongside the probe KV and maintained through the utility-driven eviction described in Section 3.3.

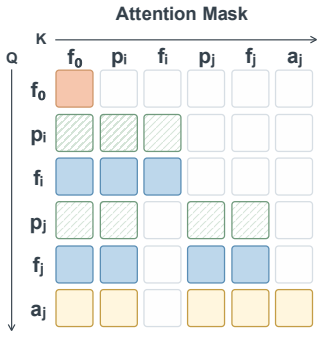


Figure 3: **Causal mask for probe aggregation.** Colored cells denote allowed attention. Here, f_0 denotes the anchor tokens; f_i, f_j are clean visual tokens; p_i, p_j are probe tokens; and a_j denotes action tokens, with $0 < i < j$.

To preserve fine-grained visual evidence, we retain the original historical visual KV states and use probes as dynamic routers. Similarities between probe keys and queries in DiTs produce frame-level scores for temporal Top-K_1 sparse routing. Attention then combines their original visual KV for fine-grained details with probe KV for coarse-grained temporal context, while the probes also assess redundancy during KV Set maintenance. Since initial probes are content-agnostic, routing is disabled in the first Transformer block ($\ell = 0$) and activated from the second ($\ell = 1$). Frame-level scoring and hierarchical block selection are detailed in Section 3.4.

3.3 Fixed-Capacity Key–Value Set via Utility-Driven Eviction

While probe tokens enable compact historical relevance estimation, sparse access alone does not bound the accumulation of historical KV states. Retaining the entire history still causes cache storage and the routing candidate space to grow with the interaction horizon. We therefore introduce a Fixed KV Set that combines recursive probe representations with utility-driven eviction to maintain temporally extended context under a bounded storage budget.

Decoupled Memory and Bounded Storage. As illustrated in Figure 2(b), the attention context decouples the permanent anchor set $\mathcal{M}_{\text{anc}}$, an unconditionally retained local recent window $\mathcal{R}_t$, and a bounded layer-specific Fixed KV Set $\mathcal{S}^{\ell}_{t}$ with capacity C_{KV}. The active context at layer ℓ is $\mathcal{C}^{\ell}_{\text{attn},t} = \mathcal{M}_{\text{anc}} \cup \mathcal{R}_t \cup \text{TopK}_{f \in \mathcal{S}^{\ell}_{t}}(s^{\ell}_{f}, K_1)$, restricting sparse retrieval to $\mathcal{S}^{\ell}_{t}$. When a frame leaves $\mathcal{R}_t$, each routing-enabled layer independently admits or replaces it.

Time-Decayed Utility via Probation Scoring. To prevent premature eviction of newly transferred frames, we decouple attention reading from eviction utility scoring. While each historical router retrieves only from $\mathcal{S}^{\ell}_{t}$, an auxiliary manager Top-K_1 scores candidates across the joint pool $\mathcal{S}^{\ell}_{t} \cup \mathcal{R}_t$. For entry r at layer ℓ, $h^{(t)}_{r,\ell}$ equally combines the single clean-video route with action-route usage averaged across denoising steps after within-layer head aggregation. Its utility follows an EMA:

$$u^{(t)}_{r,\ell} = \beta u^{(t-1)}_{r,\ell} + (1-\beta) h^{(t)}_{r,\ell}, \quad 0 < \beta < 1. \tag{4}$$

By participating in manager scoring while still residing in $\mathcal{R}_t$, incoming frames accumulate *probation utility*, avoiding cold-start bias during fixed-set replacement.

Redundancy-Constrained Eviction. Evicting entries purely by lowest utility risks discarding rarely accessed states that preserve unique historical context. At layer ℓ, let $\mathcal{U}^{\ell}_{t} = \mathcal{S}^{\ell}_{t-1} \cup \{f_{\text{in}}\}$ denote the update candidates. We identify their most redundant pair using that layer's probe descriptors and evict its lower-utility member

$$(i^{\star}_{\ell}, j^{\star}_{\ell}) = \underset{\substack{i,j \in \mathcal{U}^{\ell}_{t} \\ i \neq j}}{\arg\max}\ \text{sim}(\mathbf{P}^{\ell}_{i}, \mathbf{P}^{\ell}_{j}), \quad e^{\star}_{\ell} = \underset{k \in \{i^{\star}_{\ell}, j^{\star}_{\ell}\}}{\arg\min}\ u^{(t-1)}_{k,\ell}. \tag{5}$$

where sim denotes cosine similarity between flattened probes $\mathbf{P}^{\ell}_{i}, \mathbf{P}^{\ell}_{j} \in \mathbb{R}^{N_p \times d}$, serving as low-cost contextual compressions that retain rich semantics, especially across visually similar frames. Figure 2(b) shows that the arrival of entry D renders A and B the most redundant pair, resulting in the eviction of the lower-utility entry B and yielding the updated set $\{\text{A}, \text{C}, \text{D}\}$. Crucially, entry

C survives despite having lower utility than B, as its distinct semantics preserve diversity under a bounded cache budget.

3.4 Sparse Hierarchical Attention

While the Fixed KV Set bounds historical storage, current predictions need not access every retained visual token. We instantiate the coarse-to-fine hierarchy within MoT self-attention, before the subsequent cross-attention stage: probe-based routing first identifies relevant historical frames, and camera-aware spatial partitioning then extracts critical visual regions for fine-grained attention.

Probe-Driven Temporal Routing. Scoring all historical raw keys incurs substantial computational overhead. In long-context LLMs and video generation models, parameter-free mean pooling is commonly used for lightweight KV routing (Lu et al., 2025; Xu et al., 2026; Wu et al., 2026), but is inadequate for robotic manipulation. Cross-camera averaging mixes distinct projections, while whole-view pooling dilutes localized contact cues within background content. We instead employ learnable probes as compact and content-adaptive routing interfaces.

For a single routing event at layer ℓ, we suppress control- and denoising-step indices. For each head $z \in \{1, \dots, H\}$, let $\mathbf{Q}_z^\ell \in \mathbb{R}^{N_q \times d_h}$ denote the current query matrix, with N_q query tokens from the routed branch, and let $\mathbf{K}_z^{P,\ell}$ concatenate candidate probe keys, with N_p probes per frame. We score each historical frame by probe attention averaged across queries and heads:

$$s_f^\ell = \frac{1}{HN_q} \sum_{z=1}^{H} \sum_{q=1}^{N_q} \sum_{p=1}^{N_p} \left[\mathrm{Softmax}\left(\frac{\mathbf{Q}_z^\ell (\mathbf{K}_z^{P,\ell})^\top}{\sqrt{d_h}} \right) \right]_{q,(f,p)}. \tag{6}$$

where softmax is normalized over all candidate probe keys, and the highest-scoring indices are routed by

$$\mathcal{F}^\ell = \mathrm{TopK}_{f \in \mathcal{S}_t^\ell}(s_f^\ell, K_1). \tag{7}$$

Thus, all heads within a layer share its selected frame IDs, while different layers route independently. The default temporal routing budget is reported in Table 6 and varied in the ablation study. Recursive aggregation enables probes to carry temporal context beyond their originating frames, supporting compact temporal routing while preserving selected raw KV states for subsequent spatial partitioning.

Camera-Aware Spatial Partitioning. While temporal routing identifies relevant frames via probe tokens, decision-critical evidence within them is often localized. We therefore partition the raw KV states of each routed frame into contiguous blocks that do not cross camera boundaries. We also partition the N_q rows of $\mathbf{Q}$ into N_g contiguous groups and mean-pool each group. Let $\overline{\mathbf{Q}} \in \mathbb{R}^{N_g \times d_h}$ contain these query-group representatives, and let $\overline{\mathbf{K}}_f \in \mathbb{R}^{N_f^{\mathrm{blk}} \times d_h}$ contain the mean-pooled raw keys within each of frame f's N_f^{blk} spatial blocks. The relevance of block b is

$$\rho_{f,b} = \frac{1}{N_g} \sum_{g=1}^{N_g} \left[\mathrm{Softmax}\left(\frac{\overline{\mathbf{Q}}\, \overline{\mathbf{K}}_f^\top}{\sqrt{d_h}} \right) \right]_{g,b}. \tag{8}$$

Let $b \in \{1, \dots, N_f^{\mathrm{blk}}\}$ index the partitioned spatial blocks, $\mathcal{J}$ denote the set of camera views, $\mathcal{I}_{f,c}$ denote the blocks belonging to view c, and $K_{2,c}$ its spatial budget. The selected block-index set is

$$\mathcal{B}_f = \bigcup_{c \in \mathcal{J}} \mathrm{TopK}_{b \in \mathcal{I}_{f,c}}(\rho_{f,b}, K_{2,c}), \qquad \sum_{c \in \mathcal{J}} K_{2,c} = K_2. \tag{9}$$

This per-camera selection prevents cross-view mixing. In bimanual settings, we impose $K_{2,\mathrm{left}} = K_{2,\mathrm{right}}$ to preserve access to both wrists even during single-arm-dominant phases.

Bounded Hybrid Context and Inference Cost. As illustrated in Figure 2(c), the bounded hybrid context combines dense anchor tokens from the initial two frames, unconditionally retained recent tokens, and routed historical states. Dense anchor attention preserves lossless access to task goals. Under fixed K_1 and K_2 budgets and the settings in Appendix A.1, historical attention and cache management are $O(1)$ per decision in trajectory length.

4 EXPERIMENTS

4.1 TRAINING OBJECTIVES AND IMPLEMENTATION DETAILS

Training Objectives. Following Fast-WAM (Yuan et al., 2026), we adopt visual dynamics prediction only during training to ground physical representations, omitting explicit pixel generation at inference to preserve inference efficiency. We adopt its backbone initialization and joint flow-matching objectives. To supervise discrete selection, we augment optimization with an auxiliary routing loss, yielding the overall objective

$$\mathcal{L} = \mathcal{L}_{\text{video}} + \mathcal{L}_{\text{action}} + \lambda_{\text{r}}\mathcal{L}_{\text{route}}, \tag{10}$$

where $\mathcal{L}_{\text{video}}$ and $\mathcal{L}_{\text{action}}$ are timestep-weighted velocity prediction losses for visual dynamics and actions, respectively, and λ_{r} weights routing supervision.

Surrogate Routing Supervision. Because discrete Top-K_1 routing is non-differentiable, gradients cannot directly supervise probe routing. We therefore guide probe scores using a detached dense-attention teacher normalized over all valid keys before sparse routing. Summing its attention within each candidate frame and averaging across heads yields frame-level target masses $m_{g,f}$. We minimize the target-weighted Kullback–Leibler divergence (Kullback & Leibler, 1951)

$$\mathcal{L}_{\text{route}} = \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}} Z_g\, D_{\text{KL}}(\mathbf{t}_g \| \mathbf{p}_g), \tag{11}$$

where $Z_g = \sum_f m_{g,f}$ is the historical attention mass, $\mathbf{t}_g = [m_{g,f}/Z_g]_f$ is the normalized teacher distribution, and $\mathbf{p}_g$ is the probe distribution. Query groups with $Z_g \geq 10^{-6}$ form $\mathcal{G}$. Both Z_g and $\mathbf{t}_g$ are stop-gradient targets. The KL divergence trains relative probe ranking, while Z_g discounts query groups that barely attend to history. To ensure training efficiency without dense overhead, this teacher supervision is sampled stochastically with probability $p_{\text{sup}} = 0.2$ and discarded at inference.

Two-Stage Training Scheme. Simultaneously training visual dynamics, probe representations, and dynamic cache eviction from scratch introduces severe optimization non-stationarity, because untrained probe tokens produce erratic routing and arbitrary memory eviction during early iterations. To ensure stable convergence, we introduce a progressive two-stage training scheme. In Stage 1, we disable cache eviction, retaining all historical frames alongside their probe tokens and raw key-value states. This stationary regime allows learnable probes to steadily master multi-camera observation compression, recursive causal aggregation, and router calibration without memory loss. In Stage 2, we introduce the Fixed KV Set and activate online utility-driven eviction, enabling the network to adapt smoothly to bounded-context memory management and robust long-horizon execution.

Implementation Details. Training runs on four NVIDIA B300 GPUs with a per-GPU batch size of four and a learning rate of 10^{-4}. Additional implementation details are provided in Appendix A.1.

4.2 SIMULATION BENCHMARK RESULTS

Benchmarks and Baselines. We evaluate long-horizon robotic manipulation on the memory-dependent RMBench (Chen et al., 2026) and LIBERO-Mem (Chung et al., 2026), as well as the general-purpose LIBERO suites (Liu et al., 2023).

Evaluation on RMBench. As shown in Table 1, SHARP-WAM establishes a state-of-the-art with a 76.0% average success rate across memory-intensive manipulation, outperforming memoryless policies ($\pi_{0.5}$, Fast-WAM), and memory-augmented baselines including Mem-0 and HiMem-WAM. Crucially, on sequential $M(n)$ tasks, SHARP-WAM surpasses LingBot-VA by 8.8 percentage points despite the latter's robot-data pretraining. Evaluation and baseline are detailed in Appendix B.2.

Evaluation on LIBERO-Mem. Table 2 shows that SHARP-WAM achieves the best average success rate of 47.0%, outperforming the strongest baseline, MemoryVAM, by 4.5 percentage points. SHARP-WAM ranks first on five of the ten tasks and delivers particularly large gains on long-horizon repeated-operation tasks, improving Bt$\times$3 from 10.0% to 80.0% and B$\times$7 from 0.0% to 30.0%. With video-prediction supervision and effective long-horizon memory, SHARP-WAM performs strongly on repeated-operation, sequencing, and spatial-reasoning tasks.

Evaluation on LIBERO. Table 3 shows that SHARP-WAM achieves a 97.3% success rate, surpassing general-purpose models such as GR00T-N1 and $\pi_{0.5}$, as well as memory-bank methods includ-

Table 1: **Success rates (%) on RMBench.** Each task uses 50 expert demonstrations for training and 100 rollouts for evaluation. **Bold** and underlined values indicate the best and second-best results.

	$M(1)$ tasks						$M(n)$ tasks					
Method	Observe & Pick Up	Rearrange Blocks	Put Back Block	Swap Blocks	Swap T	Avg.	Battery Try	Blocks Ranking Try	Cover Blocks	Press Button	Avg.	Overall Avg.
Visuomotor Policies												
DP (Chi et al., 2025)	1	0	0	11	20	6.4	10	10	0	0	5.0	5.8
ACT (Zhao et al., 2023)	1	29	0	2	2	6.8	19	0	0	0	4.8	5.9
Vision-Language-Action Models												
$\pi_{0.5}$ (Black et al., 2025a)	9	13	11	24	15	14.4	16	6	0	0	5.5	10.4
X-VLA (Zheng et al., 2026)	9	13	18	16	3	11.8	26	1	2	0	7.3	9.8
Mem-0 (Chen et al., 2026)	4	89	90	67	14	52.8	28	18	68	0	28.5	42.0
World-Action Models												
Fast-WAM (Yuan et al., 2026)	0	0	0	0	13	2.6	16	37	0	0	13.3	7.3
LingBot-VA (Li et al., 2026a)	14	99	**100**	77	**94**	**76.8**	32	**100**	83	67	70.5	74.0
WLA-0 (Yang et al., 2026b)	–	–	–	–	–	–	**45**	23	84	74	56.5	–
HiMem-WAM (Sun et al., 2026)	**28**	33	32	38	27	31.6	28	24	19	8	19.8	26.3
SHARP-WAM (Ours)	6	**100**	97	**84**	80	73.4	**45**	**100**	**90**	**82**	**79.3**	**76.0**

Table 2: **Success rates (%) on LIBERO-Mem.** Each task is evaluated over 20 rollouts. Abbreviations: B=Bowl, Bt=Bottle, P=Plate, Cs=Cheese, Bskt=Basket, and E=Empty basket.

	Single		Repetition				Sequencing / Spatial				
Method	B→P	Bt→P	B×3	Bt×3	B×5	B×7	Swap2	Rot3	Cs→Bskt	Cs+E	Avg.
π_0 (Black et al., 2025b)	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0
SlotVLA$_{h=8}$ (Chung et al., 2026)	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0
MemoryVLA (Shi et al., 2026b)	55.0	0.0	35.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	9.5
Gated Memory Policy (Gao et al., 2026)	85.0	45.0	20.0	5.0	**40.0**	0.0	0.0	0.0	0.0	0.0	19.5
MemoryVAM$_{\text{UNet}}$ (Jiang et al., 2026)	85.0	**65.0**	**90.0**	10.0	35.0	0.0	**25.0**	5.0	40.0	**70.0**	42.5
SHARP-WAM (Ours)	**90.0**	30.0	70.0	**80.0**	30.0	**30.0**	20.0	**15.0**	**50.0**	55.0	**47.0**

Table 3: **LIBERO comparison.** Suite success rates (%) and average.

Method	Spatial	Object	Goal	Long-10	Avg.
General-purpose baselines					
GR00T-N1 (NVIDIA et al., 2025)	94.4	97.6	93.0	90.6	93.9
$\pi_{0.5}$ (Black et al., 2025a)	**98.8**	98.2	98.0	92.4	96.9
Motus (Bi et al., 2026)	96.8	99.8	96.6	**97.6**	**97.7**
Fast-WAM (Yuan et al., 2026)	98.2	**100.0**	97.0	95.2	97.6
Memory-augmented baselines					
MemoryVLA (Shi et al., 2026b)	98.4	98.4	96.4	93.4	96.7
MAP-VLA (Li et al., 2026b)	96.3	98.4	95.4	83.4	93.4
HiMem-WAM (Sun et al., 2026)	98.2	99.8	**98.4**	94.5	**97.7**
μVLA (Cherepanov et al., 2026)	93.0	99.4	96.6	95.8	96.2
SHARP-WAM (Ours)	98.5	99.5	97.0	94.0	97.3

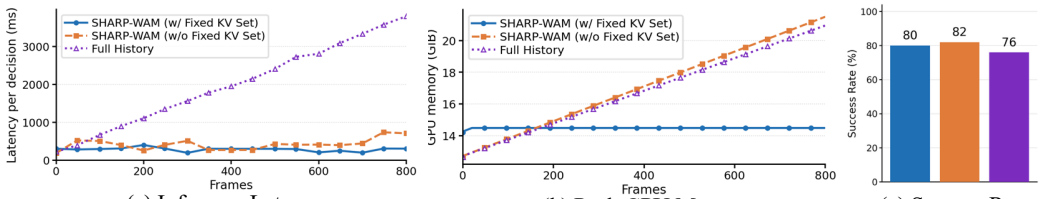
ing MemoryVLA and MAP-VLA. It also remains highly competitive with multi-stage pretraining models including Motus and HiMem-WAM. Together, these results support our backbone-native KV design for integrating historical memory while retaining general manipulation capabilities.

4.3 Real-World Experimental Results

To evaluate long-horizon manipulation in the real world, we deploy SHARP-WAM on a humanoid robotic platform performing sequential beverage reorganization. The task requires iterative pick-and-place operations through a buffer slot. All policies are adapted using the same **45** task-specific expert demonstrations and evaluated over **20** randomized trials. As shown in Figure 4, SHARP-WAM achieves a 75% success rate (15/20), outperforming LingBot-VA by 25 percentage points, while $\pi_{0.5}$ and Fast-WAM fail all trials. This margin demonstrates stronger retention of intermediate task states and more reliable multi-stage execution.



Figure 4: **Real-world task progression and success rates.** 45 expert demonstrations and 20 trials.



(a) Inference Latency (b) Peak GPU Memory (c) Success Rate

Figure 5: **Inference efficiency and memory scalability.** (a) Latency and (b) GPU memory over rollout; (c) success rates of SHARP-WAM, SHARP-WAM w/o Fixed KV Set, and Full History.

4.4 Efficiency and Scalability Analysis

Unlike typical VLAs, video-based world action models suffer severe memory and latency overheads from dense spatiotemporal tokens and multi-step denoising. Figure 5 evaluates inference efficiency and memory scalability on *Swap T* across three variants: *Full History*, *SHARP-WAM w/o Fixed KV Set eviction*, and complete *SHARP-WAM*. Both unbounded baselines climb to 22 GiB memory because sparse attention only reduces computational FLOPs without preventing unevicted KV states from linearly accumulating. In contrast, SHARP-WAM maintains $O(1)$ per-decision latency and plateaus at 14.5 GiB memory. SHARP-WAM retains an 80% success rate, matching its unbounded variant (82%) and outperforming Full History (76%), where redundant raw tokens dilute task cues.

4.5 Ablation Studies

Table 4: **Ablation of contextual memory design.** Success rates (%) on two RMBench tasks.

Variant	Swap T	Press Button
w/o Bounded Capacity	82	82
w/ FIFO Eviction	67	74
w/o Anchor Tokens	75	79
w/o Raw KV States	58	64
w/o Probe Tokens	71	71
SHARP-WAM (Ours)	80	82

Table 5: **Ablation of hierarchical sparse attention.** Success rates (%) on two tasks.

Variant	Swap T	Press Button
$K_1 = 8$	77	81
$K_1 = 16$	80	82
$K_1 = 24$	72	75
$K_1 = 32$	73	75
w/o Spatial Partitioning	80	84
w/o Bimanual Symmetry	75	79

Tables 4 and 5 validate our architectural designs. The bounded cache remains within 2 points of the unbounded variant, whereas FIFO eviction incurs drops of 13 and 8 percentage points, confirming the necessity of utility-driven retention. Removing raw KV states yields the sharpest drops (22 and 18 points), while removing probes or anchors also degrades success, demonstrating their complementary roles. Accuracy peaks at $K_1 = 16$, while $K_1 \geq 24$ causes drops of 7–8 points as redundant history dilutes decision-critical cues. Finally, camera-aware partitioning preserves accuracy with fewer tokens, and bimanual symmetry prevents wrist-view imbalance.

5 Conclusion

We present SHARP-WAM, a bounded-context World Action Model unifying backbone-native historical memory with horizon-invariant $O(1)$ inference complexity. By coupling learnable probe tokens with utility-driven KV eviction and hierarchical sparse attention, SHARP-WAM achieves competitive long-horizon manipulation without external plug-ins. Future work will investigate ultra-long-horizon tasks lasting tens of minutes and test-time training for memory adaptation.

LLM Disclosure Statement

Generative AI tools were used to support experimental design and to assist with literature search, research code editing and execution, and language refinement. All AI-assisted outputs were critically reviewed and verified by the authors.

Reproducibility Statement

To support the reproducibility of our research, we provide the training objectives and implementation details of SHARP-WAM in Section 4.1, with additional hyperparameters and evaluation protocols in the appendix. To foster academic exchange and technical transparency, we will publicly release our source code, model configurations, and fine-tuned model weights in accordance with the applicable licenses through our project page. These resources will enable other researchers to reproduce our results and build upon this work.

Ethics Statement

This work is conducted solely for academic research. The capabilities and outputs discussed in this paper should not be interpreted as reflecting or endorsing the authors' views. We advocate the responsible development of AI technologies with due consideration for ethics, safety, and broader societal values.

References

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

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.

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.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, Ru Ying, Ran Zheng, and Yao Mu. AHA-WAM: Asynchronous horizon-adaptive world-action modeling with observation-guided context routing. *arXiv preprint arXiv:2606.09811*, 2026.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020*, pp. 213–229. Springer, 2020.

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

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*, 2026.

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. *The International Journal of Robotics Research*, 44(10–11):1684–1704, 2025.

Nhat Chung, Taisei Hanyu, Toan Nguyen, Huy Le, Frederick Bumgarner, Duy Minh Ho Nguyen, Khoa Vo, Kashu Yamazaki, Chase Rainwater, Tung Kieu, Anh Nguyen, and Ngan Le. Rethinking progression of memory state in robotic manipulation: An object-centric perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3407–3415, 2026.

Yihuai Gao, Jeff Jinyun Liu, Shuang Li, and Shuran Song. Gated memory policy: In-context memorization and adaptation. In *10th Annual Conference on Robot Learning*, 2026.

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

Yuxin Jiang, Chang Yu, Yunuo Chen, Xiang Feng, Yin Yang, Nishank Gite, and Chenfanfu Jiang. MemoryVAM: Integrating memory into video action model for robot manipulation. In *10th Annual Conference on Robot Learning*, 2026.

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

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-tuning video models for visuomotor control and planning. In *The Fourteenth International Conference on Learning Representations*, 2026.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Proceedings of Robotics: Science and Systems*, 2026a.

Runhao Li, Wenkai Guo, Zhenyu Wu, Changyuan Wang, Haoyuan Deng, Zhenyu Weng, Yap-Peng Tan, and Ziwei Wang. MAP-VLA: Memory-augmented prompting for vision-language-action model in robotic manipulation. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026b.

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. Datasets and Benchmarks Track.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, et al. MoBA: Mixture of block attention for long-context LLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

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

Yuzhen Mao, Michael Y. Li, and Emily B. Fox. Simplified sparse attention via gist tokens. *arXiv preprint arXiv:2604.20920*, 2026.

MotuBrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, et al. MotuBrain: An advanced world action model for robot control. *arXiv preprint arXiv:2604.27792*, 2026.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, et al. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Rutav Shah, Rajat Kumar Jenamani, Xiaohan Zhang, Lingfeng Sun, Roberto Martín-Martín, Yuke Zhu, Deva Ramanan, and Karl Schmeckpeper. Scaling short-term memory of visuomotor policies for long-horizon tasks. *arXiv preprint arXiv:2606.16178*, 2026.

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.

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 *The Fourteenth International Conference on Learning Representations*, 2026b.

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

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. In *10th Annual Conference on Robot Learning*, 2026.

Jianzong Wu, Liang Hou, Haotian Yang, Ye Tian, Pengfei Wan, Di Zhang, and Yunhai Tong. VMoBA: Mixture-of-block attention for video diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 68208–68224. PMLR, 2025.

Boxun Xu, Yuming Du, Zichang Liu, Siyu Yang, Ziyang Jiang, Siqi Yan, Rajasi Saha, Albert Pumarola, Wenchen Wang, and Peng Li. Sparse Forcing: Native trainable sparse attention for real-time autoregressive diffusion video generation. *arXiv preprint arXiv:2604.21221*, 2026.

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. In *10th Annual Conference on Robot Learning*, 2026a.

Yi Yang, Zhihong Liu, Siqi Kou, Yiyang Chen, Yanzhe Hu, Jianbo Zhou, Boyuan Zhao, Zhijie Wei, Xiao Xia, Xueqi Li, Pengfei Liu, and Zhijie Deng. World-language-action model for unified world modeling, language reasoning, and action synthesis. In *10th Annual Conference on Robot Learning*, 2026b.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, et al. World action models are zero-shot policies. In *10th Annual Conference on Robot Learning*, 2026.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23078–23097. Association for Computational Linguistics, 2025.

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.

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. *arXiv preprint arXiv:2606.23589*, 2026.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric P. Xing, and Hao Helen Zhang. Faster video diffusion with trainable sparse attention. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

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

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. In *The Fourteenth International Conference on Learning Representations*, 2026.

A Implementation and Routing Details

A.1 Implementation Details

Table 6 summarizes the shared model, memory, and optimization settings. We use 10-step Euler sampling for the flow-matching process. Latency measures end-to-end policy inference, including VAE encoding, 10-step denoising, sparse routing, and KV-cache management. The efficiency measurements in Figure 5 are collected on a single NVIDIA B300 GPU. For the efficiency and scalability comparison in Figure 5, *Full History* and *SHARP-WAM w/o Fixed KV Set* are trained separately under their respective memory configurations, rather than obtained by modifying only the inference-time configuration of the trained SHARP-WAM model.

Table 6: **Implementation hyperparameters.**

Hyperparameter	Setting
Probe tokens per observation frame N_p	8
Fixed KV Set capacity C_{KV}	36 frames
Retained anchor/recent frames	$\lvert\mathcal{M}_{\mathrm{anc}}\rvert = 2$, $\lvert\mathcal{R}_t\rvert = 4$
Temporal routing budget K_1	16 frames
Eviction decay β	0.6
Routing-loss weight λ_{r}	0.3
Routing-supervision sampling p_{sup}	0.2
Flow-matching Euler steps S	10
Optimizer	AdamW
Learning rate	10^{-4}
Weight decay / gradient clipping	10^{-2} / 1.0
Per-GPU batch size	4
Training precision	bfloat16

Complexity Scope and Assumptions. We distinguish sequence-level training or recomputation cost from incremental inference cost. For a history of length T, dense self-attention applied jointly to the complete sequence has $O(T^2)$ cost, whereas full-history inference that reuses cached historical keys and values incurs $O(T)$ historical attention for the current decision. For a fixed architecture, observation resolution, anchor and recent windows, cache capacity C_{KV}, routing budgets K_1 and K_2, and number of denoising steps, SHARP-WAM's per-decision historical attention and cache-management costs are $O(1)$ in trajectory length. In particular, candidate scoring examines a bounded pool, while redundancy-based replacement can require $O(C_{\mathrm{KV}}^2)$ pairwise comparisons; this remains constant with respect to T when C_{KV} is fixed. The total cost of an entire T-decision rollout therefore still grows as $O(T)$, and bounded capacity does not imply lossless retention of all raw states from an arbitrarily long history.

A.2 Dual-Path Routing and Frame Lifecycle Specification

To simultaneously ensure horizon-invariant inference latency and robust long-term retention, SHARP-WAM strictly decouples the inference-time attention reader from the eviction-scoring memory manager.

Frame Lifecycle and Candidate Pools. Let $\mathcal{I}_{\mathrm{anc}} = \{0, 1\}$ denote the indices of the first two observation frames, whose states form the permanent anchor set $\mathcal{M}_{\mathrm{anc}}$. Anchor frames follow a separate lifecycle: they are retained for the entire rollout and never enter either the recent window $\mathcal{R}_t$ or any Fixed KV Set $\mathcal{S}_t^{\ell}$. Consequently, anchors do not consume the capacity C_{KV} or the historical routing budget K_1. Every non-anchor observation $f \notin \mathcal{I}_{\mathrm{anc}}$ progresses through the following three-stage lifecycle:

1. **Recent Window (Probation):** Upon arrival at control step $t = f$, frame f enters the local window $\mathcal{R}_t = \{j : \max(0, t-3) \le j \le t,\ j \notin \mathcal{I}_{\mathrm{anc}}\}$. In this stage, its raw key-value representations are unconditionally retained without sparse truncation.

2. **Admission to Fixed KV Set:** At step $t = f + 4$, frame f departs $\mathcal{R}_t$ and becomes the incoming entry f_{in}. Each routing-enabled layer ℓ admits it directly if $|\mathcal{S}_{t-1}^{\ell}| < C_{\text{KV}}$; otherwise, that layer performs replacement, preserving $|\mathcal{S}_t^{\ell}| \leq C_{\text{KV}}$.
3. **Historical Retention:** In each $\mathcal{S}_t^{\ell}$, the frame remains a sparse-attention candidate until layer-specific eviction.

Decoupled Candidate Spaces. We explicitly differentiate the candidate pool used for active attention calculation from that used for cache management:

- **Reader Attention Candidate Set $\mathcal{P}_{\text{reader},t}^{\ell} = \mathcal{S}_t^{\ell}$:** Layer ℓ performs sparse historical Top-K_1 retrieval over $\mathcal{S}_t^{\ell}$. The attention context combines the anchor set $\mathcal{M}_{\text{anc}}$, the recent window $\mathcal{R}_t$, and retrieved historical states:
$$\mathcal{C}_{\text{attn},t}^{\ell} = \mathcal{M}_{\text{anc}} \cup \mathcal{R}_t \cup \text{TopK}_{f \in \mathcal{P}_{\text{reader},t}^{\ell}}(s_f^{\ell}, K_1). \tag{12}$$
Therefore, the routing budget K_1 accounts exclusively for historical retrieval and does not subsume anchor or recent frames.
- **Manager Scoring Candidate Set $\mathcal{P}_{\text{manager},t}^{\ell} = \mathcal{S}_t^{\ell} \cup \mathcal{R}_t$:** To resolve the cold-start problem where incoming frames lack interaction history upon exiting $\mathcal{R}_t$, an auxiliary scoring pass evaluates probe similarities across the joint set $\mathcal{S}_t^{\ell} \cup \mathcal{R}_t$ with budget K_1. This pass outputs selection statistics exclusively to track historical importance without changing the active reader context.

Layer-Wise Pools and Shared-Head Routes. Anchor and recent frame IDs follow shared rules, while layer ℓ maintains its own Fixed KV Set $\mathcal{S}_t^{\ell}$ and independently performs admission, replacement, and eviction. At layer 0, ordinary video and action queries disable Top-K_1 retrieval from $\mathcal{S}_t^0$ and instead attend densely to raw KV from the two anchor frames and the four-frame recent window, including the current frame. The probe-token writer instead attends to raw visual KV only from the two anchor frames and the current frame, together with the current probe tokens and retained probe KV from earlier recent frames and all resident entries in $\mathcal{S}_t^0$. Thus, non-anchor historical frames, including those in the recent window, are never accessed through raw visual KV by the probe writer. Sparse Top-K_1 retrieval of historical raw KV starts at layers $\ell \in \{1, \ldots, L-1\}$, which independently select frames from their respective pools. Within each routed layer, frame scores are averaged over query tokens and attention heads before a single Top-K_1 operation. Consequently, all heads in that layer share the selected frame IDs; there is no per-head Top-K or subsequent head-wise union. Before the pools reach capacity, synchronized admission often yields identical retained IDs across layers. After saturation, layer-specific routing utilities and descriptors allow these IDs to diverge.

Decision-Local Visual and Action Routing. Within each control step t, the Video DiT performs one routing pass using its clean visual queries. The resulting historical probe keys and packed decision-local context are then held fixed throughout action generation. The Action DiT executes S denoising steps ($S = 10$ Euler steps) and recomputes its queries and Top-K_1 route independently at every step; it neither reuses the visual route nor reruns the Video DiT. Only clean-video and action routes contribute to retention statistics; noisy-video queries are excluded.

Exact Formulation of Routing Usage. Let $s_{f,v}^{(t,\ell)}$ and $s_{f,a}^{(t,\tau,\ell)}$ denote the head-averaged manager scores for frame f from the clean-video route and action denoising step τ, respectively, over $\mathcal{P}_{\text{manager},t}^{\ell} = \mathcal{S}_t^{\ell} \cup \mathcal{R}_t$. Here $\ell \in \{1, \ldots, L-1\}$ indexes routing-enabled Transformer blocks and $\mathbb{I}[\cdot]$ is the binary indicator. We aggregate the single video route and the mean action-route usage as

$$\begin{aligned} h_{r,v}^{(t,\ell)} &= \mathbb{I}\Big[r \in \text{TopK}_{f \in \mathcal{P}_{\text{manager},t}^{\ell}}\Big(s_{f,v}^{(t,\ell)}, K_1\Big)\Big], \\ h_{r,a}^{(t,\ell)} &= \frac{1}{S}\sum_{\tau=1}^{S} \mathbb{I}\Big[r \in \text{TopK}_{f \in \mathcal{P}_{\text{manager},t}^{\ell}}\Big(s_{f,a}^{(t,\tau,\ell)}, K_1\Big)\Big], \\ h_{r,\ell}^{(t)} &= \tfrac{1}{2} h_{r,v}^{(t,\ell)} + \tfrac{1}{2} h_{r,a}^{(t,\ell)}. \end{aligned} \tag{13}$$

The corresponding routing-mass statistic uses the same mean-over-steps reduction and equal video–action fusion at each layer. Thus, the S action routes jointly receive the same total weight as the

single clean-video route, while remaining responsive to the evolving action latent and diffusion timestep. A new non-anchor frame f is initialized with $u_{f,\ell}^{(f-1)} = 0$, and Equation 4 is applied once per layer at the end of each control step. Thus, when f exits $\mathcal{R}_t$ at $t = f + 4$, it carries $u_{f,\ell}^{(f+3)}$ accumulated over steps f through $f + 3$, enabling fair competition in $\mathcal{S}_t^\ell$.

Redundancy-Constrained Replacement. When $\mathcal{S}_{t-1}^\ell$ reaches capacity, layer ℓ evaluates $\mathcal{U}_t^\ell = \mathcal{S}_{t-1}^\ell \cup \{f_{\text{in}}\}$ using cosine similarity between its flattened probe matrices $\mathbf{P}^\ell$. From the most redundant pair, it evicts the lower-utility entry:

$$(i_\ell^\star, j_\ell^\star) = \underset{\substack{i,j \in \mathcal{U}_t^\ell \\ i \neq j}}{\arg\max}\ \text{sim}(\mathbf{P}_i^\ell, \mathbf{P}_j^\ell), \quad e_\ell^\star = \underset{k \in \{i_\ell^\star, j_\ell^\star\}}{\arg\min}\ u_{k,\ell}^{(t-1)}, \quad \mathcal{S}_t^\ell = \mathcal{U}_t^\ell \setminus \{e_\ell^\star\}. \tag{14}$$

The updated Fixed KV Set is obtained by removing the evicted entry $e_\ell^\star$ from $\mathcal{U}_t^\ell$. This ensures that unique but temporarily dormant trajectories survive, preserving spatiotemporal diversity under a bounded cache budget.

A.3 Benchmark-Specific Spatial Routing Configurations

The spatial budget K_2 depends on the camera configuration and the input resolution after VAE encoding and patchification. Table 7 summarizes the benchmark-specific settings.

Table 7: **Benchmark-specific spatial routing configurations.**

Benchmark	Input configuration	DiT tokens	Tokens per block	K_2 blocks	Selected raw tokens
RMBench	T-shaped mosaic, 384×320	120	10	8	80
LIBERO / LIBERO-Mem	Two 480×640 views; 192×512 composite	96	8	10	80
Real world	Head + right wrist, 480×640 each; resize to 192×256 each; 192×512 composite	96	8	10	80

Specifically, RMBench uses a T-shaped camera-view mosaic. Its 120 latent tokens are divided into 10-token blocks, and selecting $K_2 = 8$ blocks yields $8 \times 10 = 80$ raw tokens per routed frame. For LIBERO and LIBERO-Mem, the two 480×640 camera observations are independently resized to 192×256, preserving their original aspect ratio, and concatenated horizontally into a 192×512 composite image. The causal VAE applies $16\times$ spatial downsampling to produce a 12×32 latent grid. A 2×2 spatial patchification then produces a 6×16 DiT token grid, yielding 96 tokens per frame. These tokens are divided into 8-token blocks; selecting $K_2 = 10$ blocks retains $10 \times 8 = 80$ raw tokens per routed frame.

For the real-world experiments, the head-camera and right-wrist-camera streams are each captured at 480×640 and independently resized to 192×256, preserving the original aspect ratio. The two resized views are concatenated horizontally into a 192×512 composite image. The causal VAE applies $16\times$ spatial downsampling, yielding a 12×32 latent grid, and 2×2 spatial patchification converts this grid into a 6×16 DiT token grid, for a total of 96 tokens per frame. We divide these tokens into 8-token blocks and select $K_2 = 10$ blocks, retaining $10 \times 8 = 80$ raw tokens per routed frame.

B Evaluation Protocols and Experimental Details

B.1 LIBERO-Mem Evaluation Metric

For LIBERO-Mem, all ten tasks share a single jointly trained multi-task model; we do not train a separate policy for each task. The 47.0% LIBERO-Mem result reported for SHARP-WAM is the *complete-task success rate*, not the subgoal-completion rate. A rollout is counted as successful only if the policy completes all required subgoals and reaches the final task goal within the evaluation horizon; completing only a subset of the subgoals is counted as a failure. For each task, we compute

the fraction of successful complete-task rollouts over 20 trials, and report the arithmetic mean across the ten tasks in Table 2.

B.2 RMBench Evaluation Protocol and Baseline Reproduction

Following the official protocol, all nine RMBench subtasks are trained and evaluated independently, with 100 evaluation trials conducted for each subtask. For LingBot-VA (Li et al., 2026a), we use the official open-source code and the released weights pretrained on large-scale robot data, and conduct our own fine-tuning and evaluation on RMBench; the results in Table 1 are therefore obtained from our RMBench runs. For Fast-WAM (Yuan et al., 2026), we reproduce the $M(1)$ results using its official implementation, while the $M(n)$ results follow those reported by WLA-0 (Yang et al., 2026b).

B.3 Real-World Experimental Setup and Task Details

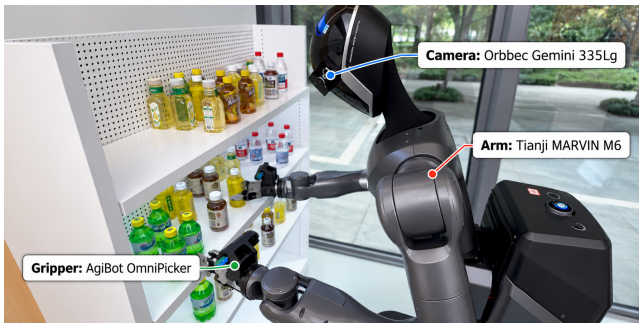


Figure 6: **Real-world platform and Sequential Beverage Reorganization task.** The humanoid robot uses synchronized observations from the head-mounted camera and right-wrist camera to execute the multi-stage rearrangement sequence.

The real-world humanoid robotic platform is shown in Figure 6. The system comprises two Tianji MARVIN M6 arms mounted on an omnidirectional mobile base, with each arm equipped with an AgiBot OmniPicker end-effector. Visual observations are captured by a head-mounted Orbbec Gemini 335Lg depth camera, which provides an egocentric global view, and eye-in-hand cameras for high-frequency local manipulation feedback. In our experiments, the policy uses only the head-mounted camera and the right-wrist camera, synchronized as a two-view visual input. Each stream is captured at a resolution of 480×640. The action space consists of continuous end-effector velocity targets and discrete gripper commands for both arms. We deploy the policy on a single NVIDIA GeForce RTX 5090 GPU and run the policy using synchronous receding-horizon execution. At each replanning cycle, the policy predicts a 32-step action chunk but dispatches only the first 16 steps to the low-level controller for execution at 5 Hz. Once the 16 dispatched steps have finished, the unexecuted final 16 predictions from the previous chunk are discarded. The newly collected observations then update both the visual context and persistent memory before the policy is queried again. Thus, the prediction horizon is 32 steps, whereas both the execution horizon and replanning interval are 16 steps. Policy inference and action execution are synchronous and do not overlap. The reported 5 Hz denotes the action-command dispatch frequency within each execution segment; each replanning cycle comprises approximately 3.2 seconds of action execution, followed by policy inference and other system overhead.

In the Sequential Beverage Reorganization task, two misplaced beverage bottles are positioned tightly side-by-side on a workspace counter, with a single vacant adjacent slot serving as a temporary buffer. The robot must complete a continuous multi-stage manipulation sequence. It first grasps the obstructing bottle and transfers it to the buffer, then picks up the second bottle and places it in its designated target slot, and finally moves the first bottle from the buffer to its matched target position. Successful completion therefore requires the policy to retain task progress across multiple dependent pick-and-place stages.

We collect only **45** expert demonstration trajectories through dual-arm teleoperation, covering diverse initial bottle placements, table textures, and ambient lighting conditions. All evaluated methods are tested on **20** randomized, unseen initial configurations to assess memory-dependent long-horizon execution. After VAE encoding and 2×2 spatial patchification, each observation contains 96 DiT tokens, following the preprocessing chain detailed in Table 7. We partition them into blocks of 8 tokens and route 10 blocks per frame, retaining $10 \times 8 = 80$ tokens for downstream DiT processing.