# SIMPLEWAM: TRAINING-FREE SINK-AWARE COMPRESSION FOR EFFICIENT WORLD ACTION MODELING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

World-action models (WAMs) jointly model future visual dynamics and robot actions, but long visual contexts and growing historical key-value (KV) caches can make inference expensive. We ask which parts of a pretrained WAM's visual context are actually necessary for action generation. Profiling action-to-visual attention in Fast-WAM and LingBot-VA reveals recurrent sink tokens with persistent action dependencies, alongside visual evidence whose relevance varies with the current action query. Based on this observation, we introduce **SimpleWAM**, a training-free sink-aware compression framework with two strategies: KeepSink protects profiled sinks while selecting the remaining action-visible tokens, and SinkAge extends sink preservation to historical KV through time-to-live scheduling and global budget allocation. On Fast-WAM, KeepSink remains competitive under a reduced token budget, whereas removing the profiled sinks drops RoboTwin Suite-10 success from 75% to 3%. SinkAge retains performance close to the Full KV while reducing the active visual-KV context by 55.1–69.4%. Experiments across WAM backbones further show that sink sensitivity varies with backbone and compression budget. In real-world experiments, KeepSink and SinkAge retain performance comparable to or better than Full KV under reduced visual-context budgets. Overall, SimpleWAM reduces active visual-KV cost without retraining while largely retaining task success, including under growing historical context.

## 1 INTRODUCTION

World-action models (WAMs) (Li et al., 2026b; Yuan et al., 2026; Ye et al., 2026; Bi et al., 2026; Kim et al., 2026) use predictive visual dynamics to support robot action generation. However, this capability can incur inference cost from dense spatiotemporal visual context, while autoregressive variants can accumulate historical KV states as the rollout horizon grows (Li et al., 2026b). Therefore, reducing inference cost becomes an important challenge for efficient WAM deployment.

Existing WAMs mainly reduce inference cost through lightweight future-imagination architectures (Li et al., 2026a), distillation of iterative video–action denoising (Akbari et al., 2026), and multi-chunk prediction for parallel rollout (Xu et al., 2026). These approaches improve efficiency by changing how the WAM is trained, structured, or executed. Visual-token pruning in VLMs and VLAs provides evidence that dense visual context contains substantial redundancy. Attention-based pruning retains visually salient tokens (Chen et al., 2024), while recent VLA methods incorporate signals more specific to embodied control: ADP combines text-conditioned relevance with end-effector dynamics (Pei et al., 2026); VLA-Pruner combines semantic-level attention with temporally estimated action-level importance (Liu et al., 2026); and VLA-IAP preserves geometric interaction anchors while adapting pruning according to semantic–motion alignment (Cheng et al., 2026). These methods demonstrate that visual context can be reduced, but also highlight that token retention depends critically on how importance is defined. Despite this progress, a complementary direction is to investigate whether pretrained WAMs already contain intrinsic visual context structures that can be exploited at inference time without changing the model. Rather than designing another importance criterion, we examine the pretrained WAM itself: *does its action pathway require access to the full visual context, or does it consistently rely on a recurrent structure within it?*

To answer this question, we first analyze action-to-visual attention in two representative WAMs, Fast-WAM (Yuan et al., 2026) and LingBot-VA (Li et al., 2026b), covering non-autoregressive and



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.

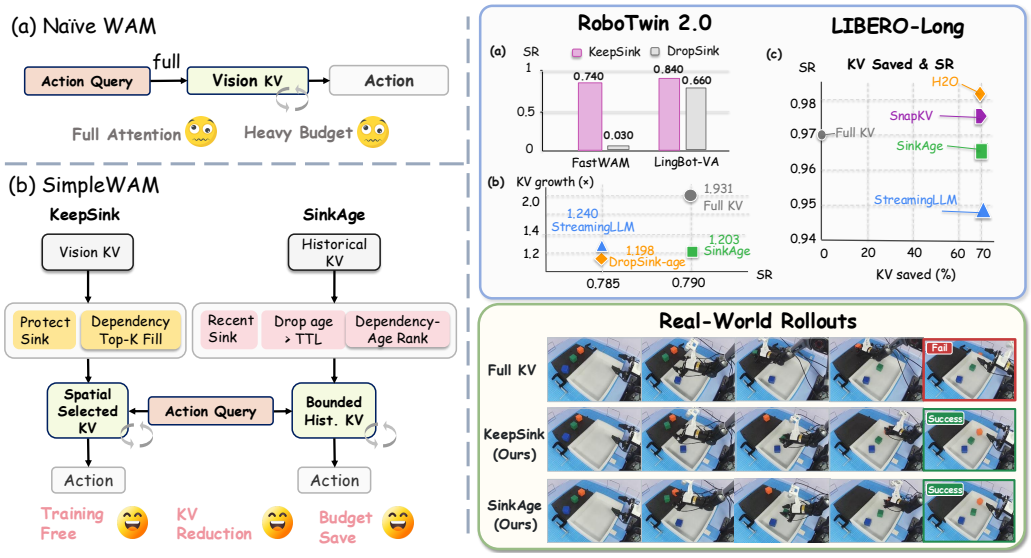



Figure 1: **Overview of SimpleWAM. Left:** Naive WAMs expose action generation to the full visual KV, whereas SimpleWAM reduces spatial redundancy with **KeepSink** and bounds growing historical KV with **SinkAge**. **Top right: (a)** KeepSink preserves higher success than DropSink under reduced visual-token budgets; **(b)** SinkAge limits historical-KV growth while retaining success compared to Full-KV; and **(c)** on LIBERO-Long, SinkAge achieves competitive success with substantial KV savings against representative training-free baselines. **Bottom right:** real-world rollouts demonstrate successful physical deployment under sink-aware compression.

autoregressive video–action modeling designs. Our profiling reveals two complementary properties of pretrained WAM visual context. **First**, action-to-visual attention exhibits recurrent positional structure across tasks and episodes: Fast-WAM consistently concentrates high attention on a single dominant visual position, whereas LingBot-VA exhibits several stable high-attention positions. **Second**, the relevance of the remaining visual tokens varies with the action query, indicating that visual context is not uniformly used during action generation.

Based on these observations, we introduce **SimpleWAM**, a training-free sink-aware compression framework that exploits recurrent visual sinks to reduce the context exposed to action generation. As illustrated in Figure 1, SimpleWAM operates along two complementary dimensions. KEEPSINK reduces spatial visual context by protecting the profiled sinks and selecting the remaining action-relevant tokens under a fixed budget, while SINKAGE extends sink-aware retention to growing historical context by bounding the active visual KV over time. Matched-budget interventions show that retaining the profiled sinks can be critical for preserving closed-loop success, with the effect varying across backbones and compression budgets. Beyond spatial compression, SINKAGE substantially reduces active historical visual KV while maintaining performance close to Full KV. Real-world experiments further show that KeepSink and SinkAge retain performance better than Full KV under reduced visual-context budgets. Specifically, our main contributions are as follows:

- **Revealing recurrent visual attention sinks in WAMs.** We identify recurrent, position-anchored visual sinks in both non-autoregressive and autoregressive backbones, and validate their functional importance through sink-preserving (KEEPSINK) and sink-removal (DropSink) interventions.
- **Training-free sink-aware compression framework.** We introduce **SimpleWAM**, a unified inference-time framework with KEEPSINK for spatial token compression and SINKAGE for bounded historical-KV allocation, requiring neither retraining nor architectural modification.
- **Cross-environment KV compression validation for closed-loop deployment.** KEEPSINK preserves Fast-WAM success with 25% fewer active visual tokens, while a single frozen SINKAGE configuration maintains near-Full performance on RoboTwin 2.0 and LIBERO-Long with 55-69% less active visual KV. Real-world experiments further show that these budget reductions remain effective under physical robot execution without retraining.

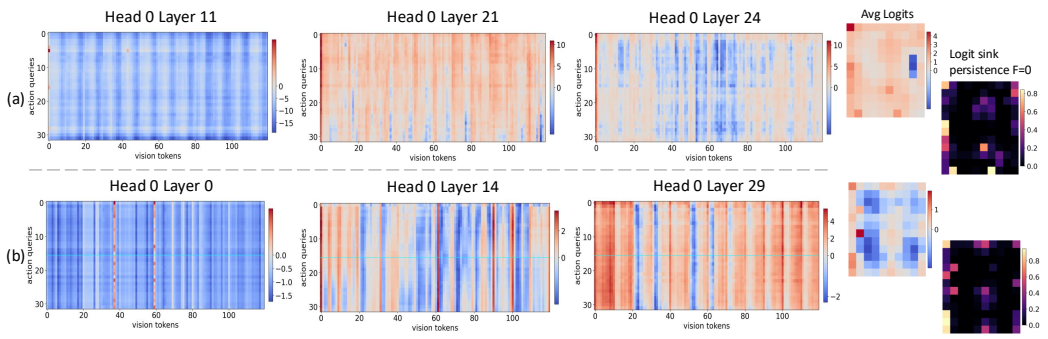



Figure 2: **Action-to-visual sink analysis in WAMs.** We visualize one `move_can_pot` episode from RoboTwin using the first observation frame. Left: action-to-visual attention logits from different Head 0 layers over 120 vision tokens. Right: episode-level average logits and sink persistence computed over all action-query-vision-token pairs, where persistence denotes the fraction of queries assigning a token to the sink set ($\geq 0.75$). (a) Fast-WAM and (b) LingBot-VA both exhibit recurrent, position-anchored visual sinks, while their distributions differ across architectures. Additional dependency and sink-position diagnostics appear in Appendix B.

## 2 Related Work

### 2.1 Attention Sinks and KV-Cache Compression in World-Action Models

Efficient long-context inference has been extensively studied in language models, where only a subset of historical KV states are often necessary for generation. Scissorhands formalizes the persistence-of-importance hypothesis and preferentially preserves previously influential states (Liu et al., 2023b). $H_2O$ retains a mixture of recent states and attention heavy hitters (Zhang et al., 2023). StreamingLLM identifies *attention sinks*: early states that receive disproportionately large attention and whose retention stabilizes streaming inference (Xiao et al., 2024). SnapKV instead estimates persistent importance from a recent observation window and compresses the prompt KV without fine-tuning (Li et al., 2024). These results establish that long context is highly redundant, but they are developed for language generation rather than closed-loop action generation.

The distinction matters for WAMs. LingBot-VA explicitly uses autoregressive visual history and KV caching (Li et al., 2026b), while Fast-WAM also couples visual dynamics with action prediction (Ye et al., 2026; Yuan et al., 2026). In this setting, visual KV is queried by action-generation states, so compression errors can directly perturb control. We therefore adapt long-context baselines through a shared physical allocator and ask which principles survive in embodied WAMs. Our analysis shows that WAM action attention contains persistent sinks as well as dynamic task-relevant evidence, but sink identity is not merely a text-prefix artifact: deleting the profiled Fast-WAM sink positions under a matched token budget can sharply reduce success in our Suite-10 intervention. This motivates sink-aware context retention specific to the action-generation pathway.

### 2.2 Training-Free Acceleration and Visual-Token Pruning for Robot Policies

Several recent methods reduce WAM cost by changing future generation itself. Efficient-WAM uses a compact video expert, sparse future latents, and asymmetric denoising; Flash-WAM applies modality-aware distillation; and Next Forcing learns multi-chunk prediction for faster convergence and parallel future generation (Li et al., 2026a; Akbari et al., 2026; Xu et al., 2026). Complementary methods exploit inference-time redundancy along different dimensions: SANTS adaptively shortens the denoising trajectory by selecting a state-dependent stopping point (Sun et al., 2026), whereas $C^3$ache reuses cached residual computation across adjacent inference chunks at corresponding denoising timesteps (Zhao et al., 2026).WorldCache instead performs training-free feature reuse inside diffusion world models based on heterogeneous token predictability (Feng et al., 2026). These methods primarily reduce the cost of future imagination or repeated intermediate computation, whereas we reduce the visual context consumed by an trained WAM action pathway.

A closely related line prunes or caches visual tokens in VLM/VLA inference. FastV and SparseVLM establish training-free visual sparsification in large vision-language models (Chen et al., 2024; Zhang et al., 2025). VLA-Cache reuses minimally changed visual KV across adjacent robot observations (Xu et al., 2025). Action-aware Dynamic Pruning adapts visual retention according to manipulation dynamics (Pei et al., 2026); VLA-Pruner combines semantic salience with action-decoding relevance (Liu et al., 2026); VLA-IAP protects interaction-critical geometric anchors (Cheng et al., 2026); and MVPruner allocates dynamic budgets across multi-view visual inputs (Yang et al., 2026). These works demonstrate that perceptual redundancy is substantial and that action relevance differs from generic visual saliency. Our focus is complementary in two ways: we study the joint world–action context structure of WAMs rather than standard VLA prefilling, and we compress both within-frame visual context and a historical KV cache whose size grows with interaction horizon.

# 3 METHODOLOGY

In this section, we introduce SimpleWAM, a training-free sink-aware context compression method for world-action modeling. We first formulate the visual-context reduction problem under a frozen pretrained WAM without modifying model architecture (Sec. 3.1). We then identify recurrent action-relevant visual anchors through action-to-visual attention profiling (Sec. 3.2), and instantiate the discovered structure with two complementary compression strategies: spatial visual token reduction with KEEPSINK (Sec. 3.3) and temporal historical-KV allocation with SINKAGE (Sec. 3.4).

## 3.1 PROBLEM SETUP

Consider a pretrained WAM that conditions action generation on language instruction $c$, current visual tokens $V_t = \{v_i\}_{i=1}^{N}$, and, for autoregressive WAMs, a historical visual KV cache $\mathcal{H}_t$. Let $A_t$ denote the action-query states used to predict the current action chunk. We do not change model parameters, training data, or the denoising objective. Our goal is to construct a smaller physical visual context $\widetilde{V}_t$ or historical cache $\widetilde{\mathcal{H}}_t$ that preserves closed-loop success.

## 3.2 ACTION-TO-VISUAL SINK PROFILING

We identify structural sink candidates once, using Algorithm 1. The design follows three observations from training-free KV compression: attention sinks can be position-anchored and visible in averaged pre-softmax attention logits (Xiao et al., 2024); a small subset of states can repeatedly dominate attention and remain important across steps (Zhang et al., 2023; Liu et al., 2023b); and useful KV selection can be estimated from a compact observation window while remaining query dependent (Li et al., 2024; Tang et al., 2024). We therefore profile only action-to-visual interactions, using masked pre-softmax logits for structural sink discovery and reserving normalized attention probabilities for the online action-conditioned dependency used later by the allocator. The profiling hyperparameters are fixed diagnostics rather than deployment budgets: we use $\rho$=0.10, $\tau_p$=0.75, $\tau_u$=0.50, and $\tau_s$=0.50, while the deployment keep ratio $\kappa$ and historical budget $B$ are evaluated separately. For deployment, $m$ controls the number of profiled sink candidates frozen in $S$, ranked by $s_v$. Figure 2 visualizes sink profiles for Fast-WAM and LingBot-VA, including layer-wise action-to-visual attention logits and episode-level average-logit and persistence maps that reveal recurrent visual sink patterns.

**Backbone-specific visual segments.** For Fast-WAM, $\mathcal{V}$ contains the 120 first-frame visual tokens visible to action queries. For LingBot-VA, we profile only the visual observation tokens `video_obs[0:120]` as the candidate visual context, while keeping action-observation and temporary denoising-action keys outside the visual-token compression scope. This action-pathway restriction follows the query-conditioned principle of query-aware KV selection (Tang et al., 2024) and is consistent with action-aware visual pruning results showing that generic visual salience need not coincide with action relevance (Pei et al., 2026; Liu et al., 2026).

**Action-conditioned dependency.** After the sink set is frozen, the remaining visual budget is allocated using normalized action-to-visual attention probabilities. We summarize the dependency of visual token $i$ as:

$$d_i = \mathrm{Agg}\Big(\{\alpha_{q\to i}^{(\ell,t,h)} : q \in A_t\}\Big), \tag{1}$$

**Algorithm 1** Training-free sink-token profiling

**Require:** Pretrained WAM $b$, profiling rollouts $\mathcal{D}$, top fraction $\rho$, thresholds $\tau_p, \tau_u, \tau_s$, optional deployment cardinality $m$
**Ensure:** Sink statistics $\{\mu_v, p_v, s_v\}$, existence flag $E$, and frozen sink set $S$

$\mathcal{V} \leftarrow$ CANONICALVISUALKEYS$(b)$; $\mathcal{C} \leftarrow \varnothing$
**for all** $e \in \mathcal{D}$ **do**
  **for all** denoising steps $t$ and layers $\ell$ **do**
    $Z \leftarrow$ MASKEDPRESOFTMAXLOGITS$(b, e, t, \ell)$
    $G_{e,t,\ell,v} \leftarrow \mathrm{nanmean}_{h,q}\, Z_{h,q,v}$ for all $v \in \mathcal{V}$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(e, t, \ell)\}$
  **end for**
**end for**
$k \leftarrow \max(1, \lceil \rho |\mathcal{V}| \rceil)$
**for all** $v \in \mathcal{V}$ **do**
  $\mu_v \leftarrow \mathrm{nanmean}_{c \in \mathcal{C}}\, G_{c,v}$
  $p_v \leftarrow |\mathcal{C}|^{-1} \sum_{c \in \mathcal{C}} \mathbf{1}[v \in \mathrm{TopK}(G_{c,:}, k)]$
**end for**
$u_v \leftarrow \mathrm{MinMax}(\mu_v)$; $s_v \leftarrow u_v p_v$
$S^{\mathrm{strong}} \leftarrow \{v : p_v \geq \tau_p \wedge u_v \geq \tau_u\}$
$v^\star \leftarrow \arg\max_v s_v$; $E \leftarrow \mathbf{1}[p_{v^\star} \geq \tau_p \wedge s_{v^\star} \geq \tau_s]$
**if** $m$ is specified **then**
  $S \leftarrow \mathrm{TopM}_v(s_v, m)$
**else**
  $S \leftarrow S^{\mathrm{strong}}$
**end if**
**return** $\{\mu_v, p_v, s_v\}, E, S$

where the aggregation follows each backbone's native attention layout. This online signal is used only for allocation; it is kept separate from the structural sink profile.

### 3.3 SPATIAL COMPRESSION WITH KEEPSINK

For a fixed visual context of $N$ tokens, KEEPSINK protects every sink and spends the remaining budget on non-sink tokens with the highest action-conditioned dependency. This "protected anchors + budgeted important states" structure is intentionally aligned with training-free KV policies that preserve sinks or heavy hitters while bounding cache size (Xiao et al., 2024; Zhang et al., 2023; Liu et al., 2023b); our difference is that the residual budget is scored by the WAM's action pathway rather than by text-generation attention. Given target keep ratio $\kappa$ and $N_{\mathrm{act}} = \lfloor \kappa N \rfloor$, the retained set is

$$\mathcal{K}_{\mathrm{spatial}} = S \cup \mathrm{TopK}_{i \notin S}(d_i,\, N_{\mathrm{act}} - |S|)\,. \tag{2}$$

The corresponding key and value tensors are *physically gathered* at the action phase, so the active attention length is truly $N_{\mathrm{act}}$ rather than an unchanged tensor with a binary mask. This distinction makes active-token and latency reductions measurable rather than nominal.

### 3.4 TEMPORAL COMPRESSION WITH SINKAGE

For growing history, each historical token $i$ has age $a_i$. SINKAGE uses a recent-protection horizon $R$, mid-age horizon $M$, and hard time-to-live TTL. The design follows two well-established KV-cache observations: recent context remains broadly useful, while a sparse set of historically important states should survive beyond a pure sliding window (Xiao et al., 2024; Zhang et al., 2023); observation-window methods such as SnapKV further motivate making selection from a compact recent probe rather than repeatedly rescoring the full history (Li et al., 2024). Recent sink tokens are mandatory,

$$\mathcal{S}_R = \{i \in S : a_i \leq R\}, \tag{3}$$

and tokens with $a_i > \mathrm{TTL}$ are excluded. Candidates are ranked using dependency and age as:

$$u_i = d_i\, g_\tau(a_i) + b_{\mathrm{mid}}\, \mathbf{1}[i \in S \wedge R < a_i \leq M], \tag{4}$$

where $g_\tau$ is the implementation's age-weighting function, $\tau$ controls its age scale, and $b_{\text{mid}}$ is a mid-age sink bonus. We retain $\mathcal{S}_R$ first, fill a global budget $B$ with the highest-scoring eligible states, and physically gather the resulting KV tensors. The frozen configuration used throughout the temporal study is $H$=72, $B$=384, $R/M/\text{TTL}$=2/6/12, $\tau$=3, and $b_{\text{mid}}$=0.5.

# 4 EXPERIMENTS

In this section, we evaluate whether sink-aware context compression can reduce active visual KV while preserving closed-loop manipulation performance. We first describe the experimental settings and evaluation protocols across different WAM backbones and benchmarks (Sec. 4.1). We then analyze spatial compression with KEEPSINK and sink-removal interventions (Sec. 4.2), followed by historical KV compression with SINKAGE under long-context settings. We provide analysis to understand the effect of sink preservation and context allocation strategies (Sec. 4.3) and validate the deployment of SimpleWAM on real-world manipulation tasks (Sec. 4.4).

## 4.1 EXPERIMENTAL SETUP

**Datasets.** We evaluate on RoboTwin 2.0 (Chen et al., 2025) and LIBERO (Liu et al., 2023a). The spatial study uses Fast-WAM on 10 tasks from RoboTwin2.0 (Suite-10) with $N$=120 visual tokens and 10 evaluation episodes per task under the clean/unseen setting. The temporal study uses LingBot-VA with history horizon $H$=72 and the single frozen SINKAGE configuration above. RoboTwin temporal evaluations use 20 episodes per task; LIBERO-Long uses 50 episodes per task.

**Baselines.** For spatial compression, we additionally compare with FastV (Chen et al., 2024), an attention-based visual token pruning baseline, and two training-free token selection variants. Saliency refers to the action-guided visual pruning in (Luo et al., 2026) by ranking visual tokens using action-to-vision attention scores under the same $\kappa$=0.75 budget, but without explicitly preserving sink tokens. Random uniformly samples the same number of visual tokens without using any attention-based criterion, serving as a budget-matched lower bound. For historical compression, we compare against Full KV, StreamingLLM (Xiao et al., 2024), $H_2O$ (Zhang et al., 2023), and SnapKV (Li et al., 2024) using a shared allocator interface. StreamingLLM keeps a protected recent/sink region, $H_2O$ combines recent protection with heavy-hitter filling (heavy ratio 0.2), and SnapKV uses a recent observation window with pooled dependency Top-$K$ (window 64, average-pool kernel 5). All methods use the same physical packing interface, so active-KV counts are directly comparable.

## 4.2 MAIN RESULTS

**Sink-token profiling results.** Fast-WAM identifies token 0 as the top-ranked sink candidate by $s_v$ in all 100 task–episode profiles. For LingBot-VA, averaging $s_v$ over episodes within each task, token 0 consistently achieves the highest task-level mean sink score (0.545–0.619 across the 10 Suite-10 tasks), followed by token 100 (0.484–0.524), token 90 (0.353–0.383), and token 110 (0.327–0.397). More diagnostics are provided in Appendix Sec. B.1. Unless otherwise specified, all KEEPSINK and SINKAGE main results use only the Top-1 profiled sink token, i.e., $S = \{0\}$.

**Spatial compression evaluation.** We evaluate performance using three metrics: mean success rate (SR), average active visual KV (KV ↓), and wall-clock seconds per episode where available. Table 1 provides the strongest sink-removal intervention in our study. On Fast-WAM, KEEPSINK matches Full KV at $\kappa$=0.70 and is slightly above it at $\kappa$=0.75, while explicitly removing the profiled sink position reduces mean SR from 0.730 to 0.030 at the matched budget. Random selection also degrades, whereas FastV (Chen et al., 2024) and generic saliency remain competitive at this operating point. LingBot-VA shows a different, budget-dependent pattern: at $\kappa$=0.75, Full, KEEPSINK, and DropSink obtain comparable SRs of 0.850, 0.840, and 0.660. We attribute this contrast in part to different attention organizations across the

Table 1: Spatial compression on RoboTwin Suite-10. The best results are in bold.

| Method | $\kappa$ | SR | |
|---|---|---|---|
| | | Fast-WAM | LingBot-VA |
| Full KV | 1.00 | 0.730 | **0.850** |
| KEEPSINK | 0.75 | 0.740 | 0.840 |
| FastV | 0.75 | **0.750** | – |
| Saliency | 0.75 | 0.740 | – |
| Random | 0.75 | 0.460 | **0.850** |
| DropSink | 0.75 | 0.030 | 0.660 |

Table 2: **Historical compression at $H$=72.** Success rate and active visual-KV reduction relative to Full KV across different baselines. Best SR in each suite is bold.

| | RoboTwin ALL-50 Clean | | RoboTwin ALL-50 Randomized | | LIBERO-Long | |
|---|---|---|---|---|---|---|
| Methods | SR | KV ↓ | SR | KV ↓ | SR | KV ↓ |
| Full KV | 0.872 | – | 0.884 | – | 0.970 | – |
| SINKAGE | **0.879** | 56.2% | 0.866 | 55.1% | 0.966 | **69.4%** |
| StreamingLLM | 0.855 | 56.0% | **0.889** | 56.1% | 0.948 | 69.3% |
| $H_2O$ | 0.875 | **58.4%** | 0.875 | **57.4%** | **0.982** | **69.4%** |
| SnapKV | 0.869 | 48.8% | 0.881 | 54.0% | 0.976 | **69.4%** |

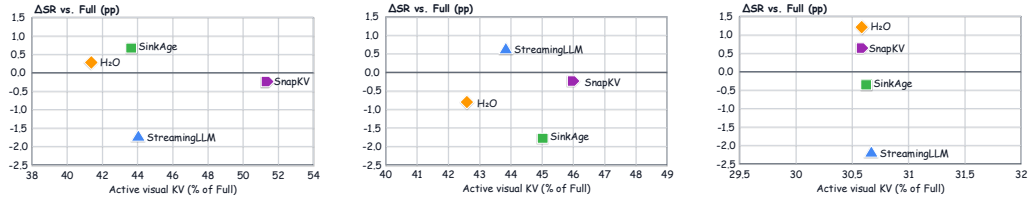
two backbones. Fast-WAM relies on a compact first-frame visual context with a highly concentrated sink, so removing that position causes a sharp collapse. LingBot-VA distributes sink-like information across autoregressive history, layers, and heads, providing greater redundancy at moderate budgets; only under tighter compression does sink removal become clearly harmful. This suggests that sink sensitivity is jointly shaped by backbone architecture and token budget. More quantitative results of KEEPSINK for LingBot-VA are reported in Appendix Sec. A.

**Historical compression evaluation.** Table 2 shows that SINKAGE reduces active visual KV while maintaining performance close to Full across all three evaluation suites. On ALL-50 `demo_clean`, its suite-mean SR is slightly above Full while removing 56.2% of active KV. Under `demo_randomized`, all five arms remain within a narrow $\pm 2$ percentage-point band of Full; StreamingLLM is slightly highest and SINKAGE is 1.8 points below Full. On LIBERO-Long, SINKAGE is within 0.4 points of Full at 69.4% active-KV reduction, while $H_2O$ and SnapKV are slightly above Full. Thus, SINKAGE substantially reduces active visual KV while maintaining performance close to Full across all three evaluation suites.

Figure 3 summarizes the success-active-KV comparison across the three benchmark suites. On Suite-10 at $H$=72, SINKAGE matches Full at 0.790 SR while reducing active KV by 65.4% (2326→806) and wall time from 204 to 128 s/episode. DropSink-age, which removes historical sink privileges while keeping the same age-aware allocator, achieves a similar 0.785 SR. The small gap between SINKAGE and DropSink-age indicates that, in this setting, bounded age-aware allocation accounts for most of the historical-compression performance, while explicit sink privilege provides only a limited additional benefit. This differs from spatial DropSink, which directly removes sink tokens from the current visual context.

Table 3: Results of Suite-10 at $H$=72, where Active KV denotes visual KV tokens exposed to action queries.

| Method | SR | Active KV | Wall s/ep |
|---|---|---|---|
| Full KV | **0.790** | 2326 | 204 |
| SINKAGE | **0.790** | 806 | 128 |
| StreamingLLM | 0.785 | **796** | **125** |
| DropSink-age | 0.785 | 809 | 129 |




(a) RoboTwin-Clean (b) RoboTwin-Random (c) LIBERO-Long

Figure 3: Quality-efficiency trade-off. Each panel plots active visual KV relative to Full against SR difference from Full for a single benchmark suite. The split view avoids mixing different suite scales and makes the near-Full quality band under large active-KV reduction easier to inspect.

### 4.3 ANALYSIS

**Action dependency shift across denoising stage.** Figure 4 compares the action-to-visual dependency change between adjacent denoising steps, averaged over 50 Suite-10 episodes (5 per task). For convenient comparison across different denoising schedules, we normalize the horizontal axis by each model's number of transitions, $k/N \in (0, 1]$. Specifically, Fast-WAM exhibits a progressive dependency shift over denoising, with the transition magnitude increasing from $6.6 \times 10^{-4}$ to

$1.33 \times 10^{-2}$ and a strictly increasing trajectory in 49/50 episodes. In contrast, LingBot-VA remains comparatively stable over most of its action-denoising trajectory.

This backbone-dependent behavior is consistent with the sink-removal results in Table 1: removing the dominant sink nearly collapses Fast-WAM performance, whereas LingBot-VA exhibits a substantial but less severe degradation under DropSink. Together, these results suggest that sink sensitivity depends on how visual dependency is organized across backbones.

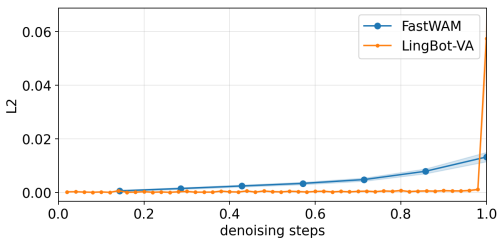



Figure 4: **Dependency shift.** Action-to-visual dependency change across denoising for Fast-WAM and LingBot-VA.

**Bounded historical computation.** Figure 5 shows that the efficiency gain of SINKAGE becomes apparent as the history horizon grows. From $H{=}12$ to $H{=}72$, Full KV increases by $1.931\times$, whereas SINKAGE grows by only $1.203\times$ under the same fixed budget and TTL. StreamingLLM and DropSink-age show similarly bounded growth of $1.240\times$ and $1.198\times$, respectively. These results indicate that SINKAGE changes the scaling behavior of the active visual context: rather than exposing action generation to a continuously growing history, it maintains a bounded active-KV budget as the rollout horizon increases.

**Sensitivity to age-aware allocation.** Table 5 in Appendix Sec. A examines the age-aware components of SINKAGE. The frozen setting with $\tau{=}3$ and a mid-age sink bonus of $0.5$ obtains 0.790 SR. Varying the age scale to $\tau{=}1.5$ and $\tau{=}6$ gives 0.775 and 0.820, respectively, while removing the mid-age bonus gives 0.800. These variations show that performance is not highly sensitive to a particular age-weighting configuration. Moreover, random filling reaches 0.845, suggesting that the exact dependency–age ranking is not necessary for competitive performance on Suite-10. These results support TTL-based age allocation for bounding historical context, while showing limited sensitivity to the specific ranking rule.

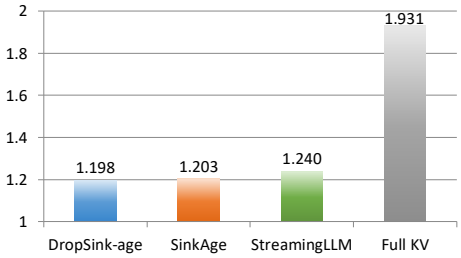



Figure 5: Active vision KV growth comparison under increasing history horizons.

### 4.4 REAL-WORLD EXPERIMENTS

We further evaluate SimpleWAM on an AgileX PiPER using LingBot-VA. The robot receives front- and wrist-view RGB observations and executes chunked control commands consisting of 6-DoF end-effector motion and 1-DoF gripper control. We consider two long-horizon manipulation tasks: *ordered three-cube color sorting* and *ordered three-fruit placement*, with randomized object configurations. Detailed hardware, control frequencies, action representation, and task protocols are provided in Appendix D. All methods use the same task-specific checkpoint and robot-control pipeline. KEEPSINK modifies only the visible visual context of action using the frozen spatial budget $\kappa{=}0.75$, while SINKAGE applies the same frozen historical-KV allocation used in the simulation. Thus, the comparison isolates the compression of the inference-time context from changes in policy parameters or low-level control. Table 4 reports 10 trials per method and task. On color sorting, Full KV, KEEPSINK, and SINKAGE achieve $20\%$, $40\%$, and $30\%$ success, respectively; on fruit placement, the corresponding rates are $40\%$, $60\%$, and $40\%$. Although the sample size is limited, KEEPSINK improves success over Full KV on both tasks under a reduced visual-token budget, while SINKAGE remains comparable to the Full-KV baseline. These results provide preliminary evidence that sink-aware visual-context

Table 4: Success rates over 10 trials per task for two real-world experiments.

| Method | Context | SR | |
|---|---|---|---|
| | | Color | Fruit |
| Full KV | Full | 20% (2/10) | 40% (4/10) |
| KEEPSINK | $\kappa{=}0.75$ | **40% (4/10)** | **60% (6/10)** |
| SINKAGE | $B{=}384$ | 30% (3/10) | 40% (4/10) |

compression can be deployed on a physical robot without retraining. The corresponding fruit-sorting trajectories are shown in Figure 6.

We point out that the real-world evaluation should be interpreted as preliminary real-world deployment results rather than as a statistically conclusive comparison. The observed outcomes are also affected by execution-level factors such as grasp instability and local-IK degradation. Nevertheless, KEEPSINK shows a consistent improvement and competitive across both tasks, while SINKAGE remains comparable to Full KV. One plausible explanation is that sink-aware compression suppresses redundant or stale visual context while retaining recurrent anchors and action-relevant evidence, thereby preserving a cleaner context for action denoising. We leave controlled verification of this mechanism to future work.

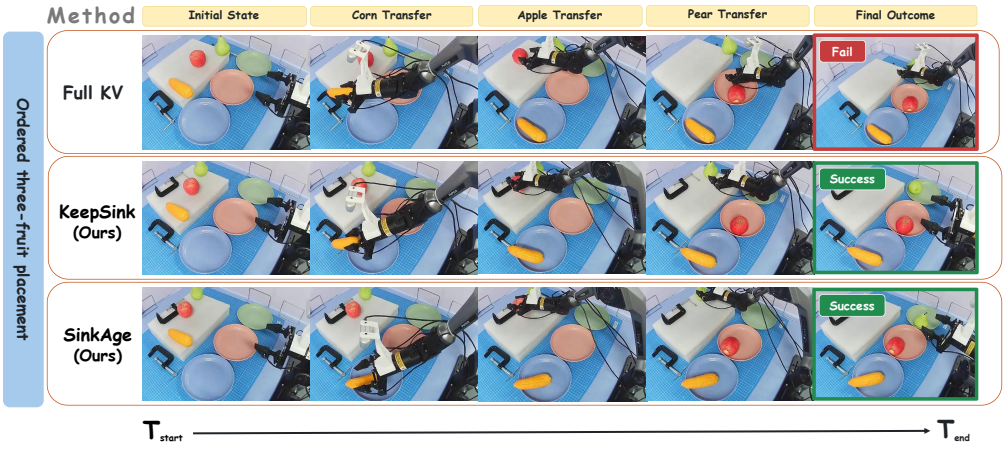



Figure 6: **Ordered three-fruit placement rollouts.** Front-camera execution sequences for the ordered three-fruit plate-placement task under Full KV, KEEPSINK, and SINKAGE.

## 5 LIMITATIONS AND FUTURE WORKS

SimpleWAM remains bounded by the capability of the underlying Full-KV policy: inference-time context optimization cannot reliably recover behaviors that are already weak under Full KV. Moreover, our current implementation reduces active visual KV without physically pruning the stored cache, so active-token savings do not yet translate proportionally into GPU-memory or end-to-end resource reductions. Finally, performance remains unstable on challenging long-horizon memory tasks; for example, in the color sorting task, after position swaps the policy may recover the correct grasp target but still fail during grasp execution, indicating that optimizing context retention alone does not resolve accumulated memory, grounding, and control errors. Future work will focus on learnable memory-budget construction, enabling the model to adaptively allocate a limited context budget.

## 6 CONCLUSION

We revisit a basic question in pretrained world-action models: does action generation require attending to the full visual context? Across two WAM backbones, we identify recurrent sink-anchored tokens within action-to-visual attention. We introduce a simple yet effective training-free token compression framework. KeepSink enables spatial context compression while largely preserving closed-loop success, and SinkAge bounds growing historical KV by retaining persistent sink anchors under a limited budget. Experimental results show that SinkAge reduces active visual KV by 55-69% while maintaining near-Full performance. Together, these results show that full visual context is not always necessary for action generation. Instead, preserving recurrent sink anchors while selectively reducing the remaining spatial and historical context can retain closed-loop performance under substantially smaller active-KV budgets. Our findings establish sink-aware token optimization to improve visual-context efficiency of WAM without retraining or architectural modification.

## AI USE STATEMENT

Generative AI tools were used to summarize and organize related literature, suggest paper structure and wording, assist with LaTeX and reference formatting, and help interpret and present experimental results supplied by the authors. They were not used to generate experimental measurements reported in this manuscript. All AI-assisted text and artifacts were reviewed by the authors, who take responsibility for the final content, claims, citations, and presentation.

## REFERENCES

Arman Akbari, Ci Zhang, Arash Akbari, Lin Zhao, Yixiao Chen, Weiwei Chen, Xuan Zhang, Geng Yuan, and Yanzhi Wang. Flash-wam: Modality-aware distillation for world action models. *arXiv preprint arXiv:2606.05254*, 2026.

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 (CVPR)*, pp. 35101–35113, June 2026.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, 2024.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Jintao Cheng, Haozhe Wang, Weibin Li, Gang Wang, Yipu Zhang, Xiaoyu Tang, Jin Wu, Xieyuanli Chen, Yunhui Liu, and Wei Zhang. Vla-iap: Training-free visual token pruning via interaction alignment for vision-language-action models. *arXiv preprint arXiv:2603.22991*, 2026.

Weilun Feng, Guoxin Fan, Haotong Qin, Chuanguang Yang, Mingqiang Wu, Yuqi Li, Xiangqi Li, Zhulin An, Libo Huang, Dingrui Wang, Longlong Liao, Michele Magno, and Yongjun Xu. Worldcache: Accelerating world models for free via heterogeneous token caching. *arXiv preprint arXiv:2603.06331*, 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.

Jiajun Li, Tiecheng Guo, Yifan Ye, Rongyu Zhang, Xiaowei Chi, Qianpu Sun, Ying Li, Yunfan Lou, Yan Huang, Zhihe Lu, Meng Guo, and Shanghang Zhang. Efficient-wam: A 1b-parameter world-action model with low-cost future imagination. *arXiv preprint arXiv:2606.10040*, 2026a.

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*, 2026b.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 22947–22970, 2024. doi: 10.52202/079017-0722.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023a.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for llm kv cache compression at test time. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52342–52364, 2023b. doi: 10.52202/075280-2279.

Ziyan Liu, Yeqiu Chen, Hongyi Cai, Tao Lin, Shuo Yang, Zheng Liu, and Bo Zhao. Bridging the semantic-action gap in visual token pruning for efficient vla inference. *arXiv preprint arXiv:2511.16449*, 2026.

Yulin Luo, Hao Chen, Zhuangzhe Wu, Bowen Sui, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Qiuxuan Feng, Jiale Yu, Shuo Gu, Peng Jia, Pheng-Ann Heng, and Shanghang Zhang. Look before acting: Enhancing vision foundation representations for vision-language-action models. *arXiv preprint arXiv:2603.15618*, 2026.

Xiaohuan Pei, Yuxing Chen, Siyu Xu, Yunke Wang, Yuheng Shi, and Chang Xu. Action-aware dynamic pruning for efficient vision-language-action manipulation. In *International Conference on Learning Representations*, 2026.

Yirui Sun, Guangyu Zhuge, Keliang Liu, Jie Gu, Shiqin Dai, Xinyu Bing, Zhongxue Gan, and Chunxu Tian. Sants: A state-adaptive scheduler for world action models. *arXiv preprint arXiv:2605.27947*, 2026.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. QUEST: Query-aware sparsity for efficient long-context LLM inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 47901–47911. PMLR, 2024.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024.

Gangwei Xu, Qihang Zhang, Jiaming Zhou, Xing Zhu, Yujun Shen, Xin Yang, and Yinghao Xu. Next forcing: Causal world modeling with multi-chunk prediction. *arXiv preprint arXiv:2606.11187*, 2026.

Siyu Xu, Yunke Wang, Chenghao Xia, Dihao Zhu, Tao Huang, and Chang Xu. Vla-cache: Efficient vision-language-action manipulation via adaptive token caching. *arXiv preprint arXiv:2502.02175*, 2025.

Nan Yang, Zhanwen Liu, Linfeng Zhang, Shangyu Xie, Yang Wang, Wenzhuo Zhou, and Xiangmo Zhao. Mvpruner: Dynamic token pruning for accelerating multi-view vision-language models in autonomous driving. *arXiv preprint arXiv:2606.27660*, 2026.

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

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

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*, 2025.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. $H_2O$: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34661–34710, 2023. doi: 10.52202/075280-1506.

Weisen Zhao, Lam Nguyen, Zhicong Lu, and Yuzhang Shang. $C^3$ache: Accelerating world action models with cross inference chunk cache. *arXiv preprint arXiv:2606.08962*, 2026.