DECONFOUNDING AND COMPRESSING HISTORY MEMORY IN STREAMING VISION-LANGUAGE-ACTION POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

How much explicit history does a streaming vision-language-action (VLA) policy need? A common test shortens its attention window, but this intervention can also shrink the physical key/value (KV) pool and change which history remains visible during denoising. We separate these quantities in LingBot-VA and show that an apparent collapse from 97.5% to 0% success on LIBERO-Long is a cache-capacity confound: prediction slots displace the previous chunk's video tokens. With the window and pool controlled, retaining the immediately preceding chunk achieves 99% success versus 97% with full history; a three-task RoboTwin replication shows the same zero-to-one retention transition. Content interventions test what the retained chunk contributes. We then combine short retention with KV quantization. On LIBERO-Long, two chunks at K4/V2 achieve 99% success, with a paired difference of 0 percentage points (95% CI: $[-3, 3]$) from the native $W = 30$ FP reference. Their history-KV storage equivalent under simulated quantization is 28.5 MiB versus 714.4 MiB of native history-slot capacity, approximately $25\times$ smaller. On the evaluated tasks, recent cached content supports high success without retaining the full past; measuring that dependence requires controlling pool capacity separately from the attention window.

1 INTRODUCTION

Streaming vision-language-action (VLA) policies generate actions one chunk at a time, and while denoising a chunk the transformer can attend to keys and values (KV) cached from the chunks before it. This cache is where the policy keeps explicit history, and its size scales with how many chunks are kept. Methods that quantize, evict, or replace cached history therefore begin from an assumption about how much of it the policy needs, and the natural test of that assumption is to shorten the attention window and watch success fall.

We show that this test can give the wrong answer. Our study uses LingBot-VA (Li et al., 2026), a streaming policy built on the Wan diffusion transformer (Team Wan et al., 2025). In the native LingBot-VA implementation, lowering the attention-window parameter from 30 to 2 reduces LIBERO-Long success from 97.5% to 0% ($n = 200$ rollouts per setting), which reads as strong evidence that the policy depends on long history. But the same parameter also sets the size of the physical KV buffer, from fifteen chunk-equivalents to one, and the denoiser writes its 128 prediction slots into that buffer. At the smallest setting they displace the video tokens of the only previous chunk the buffer can hold. Token-level tracing confirms the loss of historical video, with only intermittent historical action reads remaining (Appendix B). An action-only retention control also fails (Section 4.3). The sweep changes the attention range, the buffer capacity, and the visible history at once; the experiments below support loss of cached history under this capacity constraint as the explanation for the collapse.

To ask the question properly, we built a controlled KV store that sets three quantities independently: the attention window, the physical pool capacity, and the retention rule that decides which committed chunks stay in the pool. A six-chunk prefix comparison matches the native runtime bitwise before either configuration reaches its retention or capacity limit (Section 2). Under this store the conclusion changes. Keeping only the immediately preceding chunk gives 99% success on the ten LIBERO-Long tasks, against 97% for an uncapped reference that keeps every committed chunk, and a replication on

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.

three RoboTwin tasks shows the same jump from zero retained chunks to one. Keeping the recent chunk's slots but replacing its values does not recover this performance, so its content matters beyond cache occupancy. After a common eight-chunk warm-up, older cached content still reduces success, with a smaller gap than in the original unmatched-warm-up ladder (Section 3.2). A source-code audit of five released streaming video-diffusion systems finds the same window/pool coupling in three of them, extending the relevance of this evaluation protocol beyond the studied policy.

The corrected history requirement suggests a simple compression strategy: keep little history and store it at low precision. Against each suite's floating-point reference configuration (Section 2), four-bit keys with two-bit values (K4/V2) cost one percentage point of success on LIBERO-Long and three on RoboTwin (paired 95% CIs $[-3, 0]$ and $[-7, 0]$), whereas two-bit keys produce a clearer drop on LIBERO-Long (-6 points, CI $[-12, -1]$). Two retained chunks at K4/V2 score 99% on LIBERO-Long, a paired difference of 0 percentage points (CI $[-3, 3]$) from the native $W = 30$ floating-point reference. The two retained chunks need an estimated 28.5 MiB; the native pool's allocated history-slot capacity is 714.4 MiB. Comparing history alone gives a roughly $25\times$ storage-equivalent reduction under simulated quantization (Section 4). Reducing retention to one chunk, or keys to two bits, costs three to six points, and pruning tokens within the retained chunk trades off less favorably than lowering their precision.

Our contributions are:

1. **A controlled protocol for history-memory claims.** We separate attention window, physical KV capacity, and explicit retention, and identify prediction-slot displacement as the mechanism behind the native window-ablation collapse (Section 3.1).
2. **A characterization of the history the evaluated policy uses.** One recent chunk supports high closed-loop success on LIBERO-Long and on three RoboTwin tasks, and content interventions show sensitivity to recent value content (Sections 3.2 and 3.3).
3. **A compact operating point.** Short retention and K4/V2 compose to an estimated $25\times$ reduction in history-KV storage with no observed change in paired success, and we report where more aggressive settings begin to cost success (Section 4).

Appendix G adds weight and activation quantization results on LingBot-VA and on a second backbone, FastWAM (Yuan et al., 2026), as complementary evidence for low-bit VLA representation, and Section 6.1 reports a learned auxiliary memory that improves teacher-action prediction without a resolved gain in closed-loop success.

2 Setting and controlled evaluation

Policies and tasks. LingBot-VA (Li et al., 2026) is a streaming VLA built on the Wan diffusion transformer (Team Wan et al., 2025). It produces one action chunk at a time by denoising the chunk's tokens against a KV cache that holds the committed chunks of the current episode. On LIBERO a chunk is a stride-4 keyframe unit of 128 video tokens and 16 action tokens, 144 in total; it advances the policy by 16 control steps (the first chunk by 12), and an episode runs for at most 62 chunks. The RoboTwin configuration uses 272 tokens per chunk. The history experiments use all ten LIBERO-Long tasks (Liu et al., 2023) and, for replication, three RoboTwin tasks (Mu et al., 2025): `click_bell`, `stack_blocks_two`, and `place_dual_shoes`. The weight and activation experiments in Appendix G additionally use FastWAM (Yuan et al., 2026), a two-expert world-action model with a 5B video expert and a 1B action expert.

Three quantities the native runtime couples. We distinguish the attention window W, which bounds how far back attention may read; the physical pool capacity N_{pool}, the number of KV slots allocated; and retention, the rule that decides which committed chunks remain in the pool. The native LingBot-VA runtime derives the pool from the window,

$$N_{\text{pool}} = \lfloor W/2 \rfloor \cdot T_{\text{chunk}}, \tag{1}$$

with $T_{\text{chunk}} = 144$ on LIBERO. The denoiser writes 128 prediction slots into this shared pool, leaving capacity for at most $N_{\text{pool}} - 128$ historical tokens. At $W = 2$, this budget is too small for a complete previous chunk (Figure 1A). The token trace and the native slot-level eviction mechanism are documented in Appendix B.

Our controlled store breaks this chain. For the recent-k arms it allocates a pool independent of W and applies an explicit rule: *recent-k* keeps the k most recently committed chunks and evicts the rest. The denoising write shares this pool, so the pool is sized to hold the retained chunks and the write together: on LIBERO, 1,152 slots for $k \leq 2$ and 1,728 slots for $k = 8$. On RoboTwin the recent-k pool is 2,176 slots, exactly eight chunks, so the recent-8 arm there cannot hold eight whole previous chunks once the write is allocated; we flag this in Table 11. The *full* arm uses uncapped retention and a 65-chunk pool at $W = 130$. This covers every evaluated episode: at most 62 chunks on LIBERO and 26 on RoboTwin. Table 5 gives all pool sizes. Unless a window sweep is stated, controlled experiments use $W = 130$.

Implementation checks. The native and sparse paths produce bitwise-identical actions over a six-chunk prefix before eviction. Full-trace token replay also matches recent-1/2 retention throughout the recorded denoising forwards, including after eviction. Native $W = 8$ sparse-read reproduction exceeds the 10^{-2} action-error threshold; the native allocator supports partial-chunk states through slot-level eviction. Appendix A reports these checks, bounded reads, and lag tracing with their coverage.

Evaluation and uncertainty. Success is the binary episode outcome returned by each suite's own checker. The native LIBERO-Long sweep evaluates 10 tasks $\times$ 20 seeds ($n = 200$ per arm); controlled ladders use 10 seeds per task ($n = 100$); the original smaller staleness and content controls use 3 seeds per task ($n = 30$). On RoboTwin, two simulator-unstable seeds of `place_dual_shoes` are excluded from every arm, leaving 100 evaluated episodes per configuration (Appendix A). Arms within an experiment share task/seed units, and every paired difference resamples those units jointly (10,000 bootstrap resamples). We use same-node FP-KV references because bf16 rollouts vary between A6000 and L40S GPUs. The LIBERO retention reference is uncapped $W = 130$ (9,360 slots, 97% success); the KV-precision and composition reference is native $W = 30$ (2,160 slots, 99% success). RoboTwin KV-precision comparisons use the uncapped $W = 130$ pool (Table 2). Each paired contrast names its reference.

Memory accounting. The KV-cache results in Section 4 use simulated quantization: each committed chunk is quantized once at write time and immediately dequantized into the bf16 cache, while transient denoising slots stay in bf16 throughout. These experiments therefore measure how a low-bit representation of history affects control. The storage figures attached to them are accounting equivalents: payload bits plus one 16-bit scale and one 16-bit zero point per group. At the fine group sizes of Section 4 this metadata equals half the K4/V2 payload. The LIBERO KV reference uses the native $W = 30$ pool: 2,160 slots, or fifteen chunk-equivalents of bf16 KV at 50.6 MiB each, 759 MiB in total. Its 128 bf16 denoising slots occupy 45 MiB, leaving 714.4 MiB of allocated history-slot capacity, including slots excluded by the attention mask. We report the history-only ratio separately from the ratio including the shared denoising workspace (Section 4). Appendix E reports measured packed-weight serving memory with floating-point compute.

3 How much explicit history does the policy need?

3.1 A coupled window sweep confounds history with capacity

The native LIBERO-Long sweep gives success rates 0.975, 0.965, and 0.000 at $W = 30, 8, 2$ ($n = 200$ per arm). Equation 1 gives pools of 2,160, 576, and 144 slots: 15, 4, and 1 chunk-equivalents. Writing the 128 prediction tokens leaves capacity for 14, 3, and 0 complete historical chunks, respectively. These are capacity bounds. At $W = 8$, a cache-state census also records partial historical chunks because video and action entries are evicted separately (Appendix B).

The block-0 token trace at $W = 2$ records zero historical video tokens across 260 denoising forwards. The three passes with nonzero committed video are the first reads of the current chunk after its cache commit (Table 7). A second chunk identifier is present in only 3/260 forwards; historical action reads sum to 48 across the trace. In the controlled recent-1 trace, 150/260 forwards admit a complete committed chunk—128 video and 16 action tokens—alongside 128 prediction tokens. Controlled recent-0 admits no committed tokens in all 200 traced forwards.

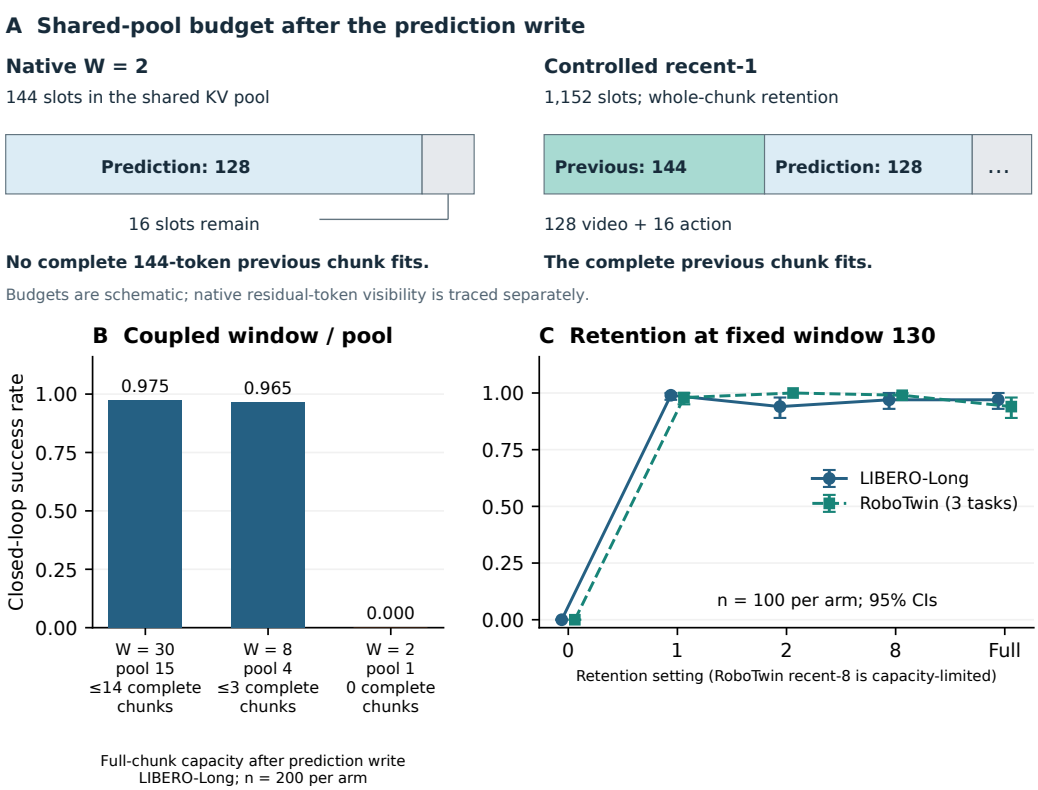

Figure 1: **Separating the window from retained history changes the conclusion.** (A) Schematic shared-pool budgets: the native $W = 2$ prediction write leaves too little capacity for a complete previous chunk; controlled recent-1 has space for retained history and the same write. Residual token visibility varies in the native trace (Appendix B). (B) Native LIBERO-Long success as both window and pool shrink ($n = 200$ per arm); labels give complete-chunk capacity bounds. (C) Controlled retention at $W = 130$ ($n = 100$ per arm; 95% bootstrap CIs). RoboTwin covers three tasks, with a capacity-limited recent-8 arm.

To separate window and retention, we run two sweeps in the controlled store. First, we fix retention at two chunks and vary W over $\{2, 8, 30, 130\}$: success is 0.97, 0.96, 0.98, and 0.94. The native collapse does not recur when capacity and retention are controlled. Then we fix $W = 130$ and vary retention: success is 0.00 with no retained chunk, 0.99 with one, 0.94 with two, and 0.97 with eight or full retention (Figure 1). Removing all retained chunks reproduces the native zero-success outcome; adding the preceding chunk restores high success. The separate action-only control also scores zero (Section 4.3). These results support cache displacement as the explanation for the native collapse.

Why can one recent chunk suffice? Its KV representation was formed while earlier chunks were available, so it may already contain information about the preceding trajectory. Retaining that representation can preserve more than the raw observation at the latest time step. The information carried through that representation is a question for future analysis (Section 6.2).

3.2 Recency and cached content matter

The retention transition leaves open whether any nonempty cache would suffice. We therefore change the age or content of the retained chunk while keeping the controlled window and cache configuration fixed (Table 1). Keeping only the first chunk of an episode scores 0.00, as does `recent-0`, whereas the same-batch `recent-1` reference scores 1.00 ($n = 30$). The original read-time lag filter changes early history availability: for $t < L$, no lag-L chunk exists and only the current chunk remains, making the first L chunks equivalent to recent-0. This code-derived warm-up behavior mixes staleness with missing early history (Appendix A).

Table 1: **Recency and content controls on LIBERO-Long.** Window 130 throughout. Lag rows use matched warm-up. The lag-filter and eviction experiments have separate references; within each experiment, arms are paired on task and seed. Mean/value replacement preserves slots, token identities, and rotary positions. Each lag arm completes all 30 episodes. Full intervals and additional controls are in Appendix B.

Intervention	Reference mechanism	n	Success [95% CI]
Immediately previous chunk (lag 1)	Read filter	30	0.933 [0.833, 1.000]
One-step-stale chunk (lag 2)	Read filter	30	0.667 [0.500, 0.833]
Three-step-stale chunk (lag 4)	Read filter	30	0.133 [0.033, 0.267]
Seven-step-stale chunk (lag 8)	Read filter	30	0.000 [0.000, 0.000]
`recent-1`, unchanged	Eviction	30	1.00 [1.00, 1.00]
First chunk only	Eviction	30	0.00 [0.00, 0.00]
Recent chunk: values replaced by means, keys intact	Eviction	30	0.00 [0.00, 0.00]

To match warm-up, all arms use the lag set $\{0, 1\}$ for chunks 0–7, then switch to $\{0, L\}$ at chunk 8. With $n = 30$ completed episodes per arm, success at lags 1, 2, 4, and 8 is 0.933, 0.667, 0.133, and 0.000. The paired lag-2 difference from lag-1 is -0.267 (95% CI $[-0.467, -0.100]$); lag-4 and lag-8 give -0.800 and -0.933, also with intervals excluding zero (Table 9). Older cached content therefore still reduces success after the common warm-up, but the lag-2 gap is smaller than in the original ladder.

Its magnitude is less than half the original -0.60 (CI $[-0.70, -0.50]$, $n = 100$). This cross-batch comparison is descriptive. On the same task–seed units, matching warm-up raises lag-2 success from 0.30 to 0.67 ($+0.37$, CI $[0.13, 0.60]$; Table 10).

We also preserve the recent chunk's slots and rotary positions while replacing cached content. Replacing keys and values by per-channel means or matched Gaussian samples gives 0.00 success. Replacing only values by their per-channel means, with every key unchanged, also gives 0.00. Success therefore depends on the recent values even when the original keys and positions are preserved. These content replacements test sensitivity to the cached representation; their distributional scope is discussed in Section 6.2.

3.3 Replication and the scope of the confound

On three RoboTwin tasks, the controlled ladder gives native window-2: 0.05, `recent-0`: 0.00, `recent-1`: 0.98, `recent-2`: 1.00, `recent-8`: 0.99, and full: 0.94 ($n = 100$ per arm). The paired `recent-1` difference from full is $+0.04$ (95% CI $[-0.01, 0.10]$). Two simulator-unstable seeds are excluded consistently across arms; details are in Appendix A. Recent-2 and recent-8 exceed full by $+0.06$ (CI $[0.02, 0.11]$) and $+0.05$ (CI $[0.01, 0.09]$), respectively (Table 11). These individual paired intervals favor recent-2 and recent-8 over full. Recent-8 is capacity-limited (Table 11). The zero-to-one retention transition replicates on these three tasks, with high success once one complete recent chunk is retained.

To assess whether window/pool coupling is an isolated implementation pattern, we audit five released streaming video-diffusion systems. Self-Forcing, LongLive, and Forcing-KV derive KV allocation and attention extent from the same configuration parameter; CausVid and Rolling Forcing use fixed allocations in the audited paths (Huang et al., 2025; Yang et al., 2026; Ji et al., 2026; Yin et al., 2025; Liu et al., 2026). The code-level audit and allocation sites are given in the appendices. This shared implementation pattern makes independent control of window, capacity, and retention relevant to a wider set of streaming systems.

4 Compressing the retained state

4.1 KV precision

With retention characterized, we quantize each committed KV write once, using per-channel key statistics over 16-token groups and per-token value statistics over 32-channel groups. Both use

Table 2: **KV precision in each suite's reference configuration.** Each arm has $n = 100$ and a same-node FP-KV reference. LIBERO uses native $W = 30$ with a 2,160-slot pool; RoboTwin uses $W = 130$ with 17,680 slots. Entries give success and paired differences with 95% bootstrap CIs. Storage remains bf16 (Section 2); per-arm intervals are in Appendix B.

Precision	LIBERO-Long: SR; paired difference	RoboTwin: SR; paired difference
FP (K16/V16)	0.99; reference	0.99; reference
K4/V4	0.98; -0.01 $[-0.05, +0.02]$	0.99; $+0.00$ $[-0.03, +0.03]$
K4/V2	0.98; -0.01 $[-0.03, 0.00]$	0.96; -0.03 $[-0.07, 0.00]$
K2/V2	0.93; -0.06 $[-0.12, -0.01]$	0.96; -0.03 $[-0.08, +0.01]$

Table 3: **LIBERO-Long KV precision and retention–precision arms.** Each arm has $n = 100$. N is the same-node native $W = 30$ FP reference; R is the controlled recent-2 FP reference. The final column reports paired differences against N, with 95% CIs. The separate contrast against R is reported in the text.

Arm	W	Pool slots	SR [95% CI]	Δ_N [95% CI]
Native FP (N)	30	2,160	0.99 [0.97, 1.00]	—
Native K4/V4	30	2,160	0.98 [0.95, 1.00]	-0.01 $[-0.05, +0.02]$
Native K4/V2	30	2,160	0.98 [0.95, 1.00]	-0.01 $[-0.03, +0.00]$
Native K2/V2	30	2,160	0.93 [0.88, 0.98]	-0.06 $[-0.12, -0.01]$
Recent-2 FP (R)	130	1,152	0.94 [0.89, 0.98]	—
Recent-2 K4/V2	130	1,152	**0.99** [0.97, 1.00]	$+0.00$ $[-0.03, +0.03]$
Recent-1 K4/V2	130	1,152	0.96 [0.92, 0.99]	-0.03 $[-0.07, +0.00]$
Recent-1 K2/V2	130	1,152	0.93 [0.88, 0.97]	-0.06 $[-0.12, -0.01]$

asymmetric min–max quantization, following the key/value asymmetry studied in KIVI (Liu et al., 2024). We leave denoising prediction slots unchanged and compare each arm with a same-node FP-KV reference at native $W = 30$ on LIBERO and $W = 130$ on RoboTwin.

K4/V4 remains close to FP on both suites (Table 2). Reducing values to two bits gives success 0.98 on LIBERO-Long and 0.96 on RoboTwin; their paired intervals allow some degradation. Reducing keys to two bits produces a clearer drop on LIBERO-Long: -0.06 with CI $[-0.12, -0.01]$. In this experiment, keys are more sensitive to the precision reduction than values. The tested asymmetric representation therefore is the one we combine with short retention below.

4.2 Retention and precision compose

Table 3 lists the native KV-precision arms and the fine-group retention–precision arms. The native $W = 30$ FP reference (N) and controlled recent-2 FP reference (R) use different windows and pools; each paired contrast below names its reference.

Two retained LIBERO chunks at K4/V2 require an estimated 28.5 MiB, including group scales and zero points. They achieve success 0.99 [0.97, 1.00] and differ from the paired native $W = 30$ FP reference on only two of 100 units, one in each direction. The paired difference is 0.00, with CI $[-0.03, 0.03]$. At saturation, the native $W = 30$ pool's allocated history-slot capacity is $(2{,}160 - 128)$ bf16 slots, or 714.4 MiB. Comparing history alone gives $714.4/28.5 \simeq 25.1$. Including the common 45 MiB bf16 prediction workspace in both totals gives $759.4/(28.5 + 45) \simeq 10.3$, using the storage-equivalent accounting defined in Section 2.

More aggressive compression reduces success (Figure 2). One chunk at K4/V2 costs 14.2 MiB, approximately $50\times$ below the same history-only reference, but scores 0.96: the paired difference from native $W = 30$ FP is -0.03 (CI $[-0.07, 0.00]$). One chunk at K2/V2 costs 11.1 MiB and scores 0.93, with difference -0.06 (CI $[-0.12, -0.01]$). On the three RoboTwin replication tasks, single-chunk K4/V2 and K2/V2 score 0.93 and 0.95 versus 0.98 for the recent-1 FP reference. Combining minimal retention with coarse precision therefore incurs an observed cost in both suites.

The recent-2 K4/V2 arm also scores above the controlled recent-2 FP reference R (0.99 versus 0.94; paired difference $+0.05$, CI $[0.00, 0.10]$). This is a separate contrast from its $+0.00$ difference against

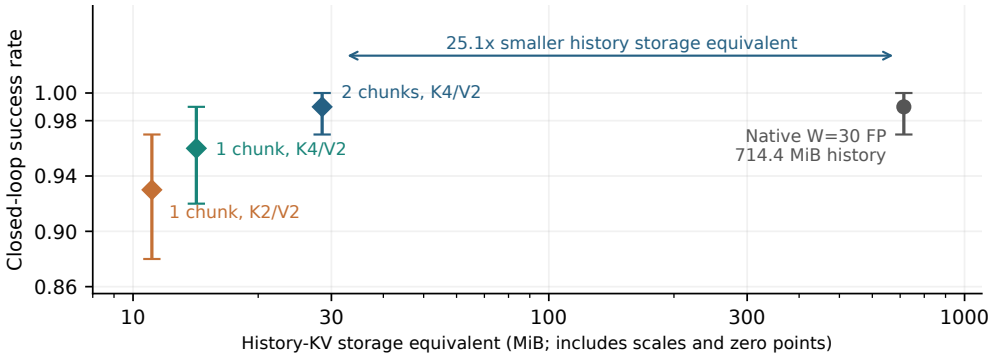

Figure 2: **History-only storage equivalents and control performance.** The native $W = 30$ FP reference is shown at its allocated history-slot capacity, 714.4 MiB, excluding the shared prediction workspace. Two chunks at K4/V2 score 0.99, the same observed success as this paired reference ($n = 100$ per arm; 95% bootstrap CIs). Low-bit points include 16-bit scales and zero points and use simulated quantization.

N. Coarser groups reduce storage to 22.7 and 20.5 MiB but score 0.94 and 0.95. Against native $W = 30$ FP, their paired differences are -0.05 (CI $[-0.10, -0.01]$) and -0.04 (CI $[-0.08, -0.01]$), respectively; both intervals exclude zero (Table 13).

We select the highest observed success among compressed KV configurations whose paired-difference CI against native $W = 30$ FP includes zero. This selects the fine-group 28.5 MiB setting, the smallest tested representation that matches the native FP reference's observed success of 0.99. This descriptive selection does not establish statistical equivalence.

4.3 Boundaries: token pruning and landmark retention

Lower precision preserves every retained token. Pruning the chunk tests a different compression axis. With recent-1 retention, uniformly reducing 144 slots to 64 gives success 0.89 and a paired difference of -0.10 (CI $[-0.16, -0.05]$, $n = 100$). Reducing to 32 or 8 slots gives 0.57 and 0.00 ($n = 30$). At 64 slots, preserving all action tokens and a contiguous group of latent tokens also gives 0.89; the paired comparison with blind striding is 0.00 (CI $[-0.08, 0.08]$). Keeping only the 16 action tokens also scores 0.00 ($n = 30$; Table 14). These tested selectors show a less favorable tradeoff than KV precision reduction.

Keeping an extra landmark chunk also provides no consistent improvement over recent-1. An attention-sink analogue scores 0.94 and a heavy-hitter analogue scores 0.97, versus recent-1 at 0.99 ($n = 100$); the configurations adapt the principles of StreamingLLM and H2O (Xiao et al., 2024; Zhang et al., 2023). The sink comparison is unfavorable, whereas the heavy-hitter difference is not resolved statistically. Full controls and intervals are in Appendix B.

Weight and activation experiments on LingBot-VA and FastWAM provide complementary evidence for low-bit representation. Their complete summary and deployment measurements are in Appendices G and E.

5 Related work

VLA policies and memory. OpenVLA, OpenVLA-OFT, and $\pi_0/\pi_{0.5}$ illustrate the broader VLA setting (Kim et al., 2025b;a; Black et al., 2025; Physical Intelligence et al., 2025). Our streaming-history study uses a Wan-based policy. VLA-Cache reuses visual computation to reduce inference cost (Xu et al., 2025), whereas our intervention concerns stored history. HiMem-WAM introduces learned memory and reports results on memory-dependent RMBench and real-world tasks (Sun et al., 2026; Chen et al., 2026). Such settings motivate testing an explicit short-history deficit before attributing control gains to long-range memory.

Quantization and cache compression. ActQuant, QuantVLA, QVLA, and QuantWAMs study low-bit VLA or world-action representations (Akbari et al., 2026; Zhang et al., 2026; Xu et al., 2026; Zhou et al., 2026). We instantiate these quantization families for comparisons within each backbone. SmoothQuant, DuQuant, and SVDQuant address weight/activation representation (Xiao et al., 2023; Lin et al., 2024; Li et al., 2025); KIVI and KVQuant address KV precision, while StreamingLLM, H2O, and VL-Cache address cache retention or token budgets (Liu et al., 2024; Hooper et al., 2024; Xiao et al., 2024; Zhang et al., 2023; Tu et al., 2025). Our contribution is to evaluate these storage choices after controlling the history requirement in closed-loop robot operation.

Streaming caches and temporal representations. The video-diffusion systems audited in Section 3 face the related choice of how much of a rolling buffer remains available after current-block writes. The auxiliary-memory study compares an explicit delay store with fixed and learned temporal representations related to HiPPO, LMU, S4, LRU, and Mamba (Gu et al., 2020; Voelker et al., 2019; Gu et al., 2022; Orvieto et al., 2023; Gu & Dao, 2024). Those comparisons measure teacher-action fidelity in the diagnostic study of Appendix D.

6 Discussion and limitations

6.1 An auxiliary-memory diagnostic: fidelity and control

We also test a quantized sparse-delay memory (QSDM) that predicts an output residual. On two RoboTwin teacher-action prediction tasks, live-policy integration reduces long-context normalized RMS from 0.607 to 0.459 (24.5%; CI $[18.3, 30.6]\%$). A separate LIBERO-Long evaluation finds no resolved success gain: recent-1 changes from 0.99 to 0.97 (paired CI $[-0.06, 0.02]$), and recent-2 from 0.94 to 0.98 (CI $[-0.01, 0.10]$), with $n = 100$ per comparison. The design, prediction experiments, and closed-loop comparisons are in Appendices C–D.

6.2 Scope and limitations

The history experiments cover one backbone, ten LIBERO-Long tasks, and three RoboTwin tasks. They measure dependence on explicitly retained KV; hidden representations may already summarize earlier processing, so a task-level memory bound remains open. The external-system audit covers source-code structure, while the FastWAM experiments concern weight and activation precision.

Token replay validates recent-1/2 retention in the recorded denoising forwards, including after eviction. Prefill, numerical KV tensors, and cross-implementation action equivalence are outside that check's scope. The native and controlled runtimes use different eviction semantics, and the RoboTwin native denoising-write width remains untraced. Content replacement can alter the feature distribution, so its effects may include distribution shift and positional interactions. High reference success limits the resolution of small performance differences; simulator and device variability remain despite pairing.

The 28.5 MiB operating point is evaluated on LIBERO-Long with simulated KV quantization. Physical packed-cache allocation, end-to-end peak VRAM, and low-bit kernel latency require deployment measurements. The auxiliary-memory study uses task-conditioned features and small in-distribution prediction sets. Teacher-action fidelity and closed-loop success are measured on different tasks, and teacher competence on the offline tasks is unestablished (Appendix F). Other trainable memories match or improve some offline QSDM results (Appendix D).

A decisive extension would establish a competent policy and a reproducible full-history advantage under the controlled protocol, then test whether compressed memory recovers that advantage against byte-matched retention and KV-precision baselines.

7 Conclusion

Controlling window, capacity, and retention reveals that the native history collapse arises from cache displacement. One recent chunk supports high success in the evaluated LingBot-VA configurations. Two chunks at K4/V2 require an estimated 28.5 MiB and score 0.99 on LIBERO-Long, matching the paired native $W = 30$ reference. This approximately $25\times$ history-only storage-equivalent reduction and the controlled protocol provide a basis for evaluating stronger history-memory claims.

REPRODUCIBILITY STATEMENT

Section 2 defines the evaluation and storage accounting; Appendix A gives pool sizes, paired sampling, exclusions, and the scope of implementation checks. Appendix B contains the trace record and per-task, precision, and pruning results. Appendices C–D detail the auxiliary-memory design and training; Appendix E gives runtime accounting and the code audit. Code, configurations, and result manifests are intended for release with the study.

AI USE STATEMENT

We used ChatGPT to critique and edit the manuscript, discuss experimental design and the interpretation and statistical reporting of results, check references, and assist with LaTeX and figure-code revisions. We also used AI tools to review our code, help implement parts of the experimental code, identify and analyze potential bugs, and investigate whether experimental results were affected by implementation errors. The revised plots use numerical summaries already reported in the manuscript. The authors take responsibility for the final text, claims, code, and figures, including material produced with AI assistance.

REFERENCES

Arash Akbari, Arman Akbari, Masih Eskandar, Qitao Tan, Yixiao Chen, Jingwu Luo, Bertha Pangaribuan, Liyun Zhang, Jennifer Dy, Geng Yuan, Xue Lin, Gaowen Liu, Stratis Ioannidis, and Yanzhi Wang. ActQuant: Sub-4-bit Action-Guided Quantization for Vision-Language-Action Models. arXiv preprint arXiv:2605.24011, 2026. URL https://arxiv.org/abs/2605.24011.

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*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Tianxing Chen, Yuran Wang, Mingleyang Li, Yan Qin, Hao Shi, Zixuan Li, Yifan Hu, Yingsheng Zhang, Kaixuan Wang, Yue Chen, Hongcheng Wang, Junjie Wang, Tianhang Yang, 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. URL https://arxiv.org/abs/2603.01229.

Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. HiPPO: Recurrent Memory with Optimal Polynomial Projections. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1474–1487. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/102f0bb6efb3a6128a3c750dd16729be-Paper.pdf.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently Modeling Long Sequences with Structured State Spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uYLFoz1vIAC.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 Million Context Length LLM Inference with KV Cache Quantization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 1270–1303. Curran Associates, Inc., 2024. doi: 10.52202/

079017-0040. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/028fcbcf85435d39a40c4d61b42c99a4-Paper-Conference.pdf.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 167283–167308. Curran Associates, Inc., 2025. doi: 10.52202/085713-5576. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f4823f831af67a3ef15e41a85434422a-Paper-Conference.pdf.

Yicheng Ji, Zhizhou Zhong, Jun Zhang, Qin Yang, XiTai Jin, Ying Qin, Wenhan Luo, Shuiyang Mao, Wei Liu, and Huan Li. Forcing-KV: Hybrid KV Cache Compression for Efficient Autoregressive Video Diffusion Models. arXiv preprint arXiv:2605.09681, 2026. URL https://arxiv.org/abs/2605.09681.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025a. doi: 10.15607/RSS.2025.XXI.017. URL https://www.roboticsproceedings.org/rss21/p017.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025b. URL https://proceedings.mlr.press/v270/kim25c.html.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal World Modeling for Robot Control. arXiv preprint arXiv:2601.21998, 2026. URL https://arxiv.org/abs/2601.21998.

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Junxian Guo, Xiuyu Li, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. SVDQuant: Absorbing Outliers by Low-Rank Component for 4-Bit Diffusion Models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vWR3KuiQur.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. DuQuant: Distributing Outliers via Dual Transformation Makes Stronger Quantized LLMs. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 87766–87800. Curran Associates, Inc., 2024. doi: 10.52202/079017-2786. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9febda1c8344cc5f2d51713964864e93-Paper-Conference.pdf.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8c3c666820ea055a77726d66fc7d447f-Paper-Datasets_and_Benchmarks.pdf.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling Forcing: Autoregressive Long Video Diffusion in Real Time. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IAyzXjbfwo.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A Tuning-Free Asymmetric 2bit Quantization for KV Cache. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32332–32344. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/liu24bz.html.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. RoboTwin: Dual-Arm Robot Benchmark with Generative Digital Twins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27649–27660, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Mu_RoboTwin_Dual-Arm_Robot_Benchmark_with_Generative_Digital_Twins_CVPR_2025_paper.html.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting Recurrent Neural Networks for Long Sequences. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 26670–26698. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/orvieto23a.html.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. arXiv preprint arXiv:2504.16054, 2025. URL https://arxiv.org/abs/2504.16054.

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. URL https://arxiv.org/abs/2606.10363.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and Advanced Large-Scale Video Generative Models. arXiv preprint arXiv:2503.20314, 2025. URL https://arxiv.org/abs/2503.20314.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. VL-Cache: Sparsity and Modality-Aware KV Cache Compression for Vision-Language Model Inference Acceleration. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HMrcv7Q4Ub.

Aaron Voelker, Ivana Kajić, and Chris Eliasmith. Legendre Memory Units: Continuous-Time Representation in Recurrent Neural Networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/952285b9b7e7a1be5aa7849f32ffff05-Paper.pdf.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38087–38099. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Siyu Xu, Yunke Wang, Chenghao Xia, Dihao Zhu, Tao Huang, and Chang Xu. VLA-Cache: Efficient Vision-Language-Action Manipulation via Adaptive Token Caching. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 164448–164473. Curran Associates, Inc., 2025. doi: 10.52202/085713-5484. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f062da1973ac9ac61fc6d44dd7fa309f-Paper-Conference.pdf.

Yuhao Xu, Yantai Yang, Zhenyang Fan, Yufan Liu, Yuming Li, Bing Li, and Zhipeng Zhang. QVLA: Not All Channels Are Equal in Vision-Language-Action Model's Quantization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=TpL2nXanru.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Ying-Cong Chen, Yao Lu, Song Han, and Yukang Chen. LongLive: Real-time Interactive Long Video Generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=nCAODkpsPJ.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From Slow Bidirectional to Fast Autoregressive Video Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yin_From_Slow_Bidirectional_to_Fast_Autoregressive_Video_Diffusion_Models_CVPR_2025_paper.html.

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. URL https://arxiv.org/abs/2603.16666.

Jingxuan Zhang, Yunta Hsieh, Zhongwei Wan, Haokun Lin, Xin Wang, Ziqi Wang, Yingtie Lei, and Mi Zhang. QuantVLA: Scale-Calibrated Post-Training Quantization for Vision-Language-Action Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39539–39549, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhang_QuantVLA_Scale-Calibrated_Post-Training_Quantization_for_Vision-Language-Action_Models_CVPR_2026_paper.html.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang "Atlas" Wang, and Beidi Chen. H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34661–34710. Curran Associates, Inc., 2023. doi: 10.52202/075280-1506. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6ceefa7b15572587b78ecfcebb2827f8-Paper-Conference.pdf.

Jiacheng Zhou, Jinfan Lv, Ruixuan Li, Longtai Zhang, Yan Wang, Wenqiang Zhang, and Lizhe Qi. QuantWAMs: Calibrating at the Right Granularity for World Action Models. arXiv preprint arXiv:2607.28405, 2026. URL https://arxiv.org/abs/2607.28405.

A EVALUATION AND IMPLEMENTATION DETAILS

Backbone and evaluation unit. LingBot-VA is a streaming Wan diffusion-transformer policy. One LIBERO chunk is a stride-4 keyframe unit containing 128 video and 16 action tokens; the RoboTwin configuration uses 272 tokens per chunk. Closed-loop success is the suite's binary episode-level outcome. The native LIBERO-Long ladder uses 10 tasks and 20 seeds per task ($n = 200$ per arm); the controlled retention and window ladders use 10 seeds per task ($n = 100$). Bootstrap intervals use 10,000 resamples over rollouts. Differences between paired arms resample the matched task–seed units jointly. Overlapping intervals or a difference interval containing zero establish neither statistical equivalence nor an upper bound on possible improvements.

Controlled interventions. The decoupled store separates attention window, physical pool capacity, and retention. LIBERO recent-k uses 1,152 slots for $k \leq 2$ and 1,728 for $k = 8$; the latter accommodates $8 \times 144 + 128 = 1,280$ retained and prediction slots. RoboTwin recent-k uses 2,176 slots, exactly eight chunks. Because prediction writes share that pool, its nominal recent-8 arm retains fewer than eight complete previous chunks. The uncapped full arms use native $W = 130$ pools: 9,360 slots on LIBERO and 17,680 on RoboTwin, both 65 chunk-equivalents. LIBERO episodes contain 10/16/62 chunks (minimum/median/maximum), so its full arm does not evict. Across all six RoboTwin arms the maximum episode length is 26 chunks, including recent-0 and native $W = 2$; the 65-chunk RoboTwin full pool therefore does not evict either. Native RoboTwin $W = 2$ uses 272 slots, one chunk. Unless a window sweep is specified, controlled retention uses $W = 130$. KV precision instead uses native $W = 30$ on LIBERO and $W = 130$ on RoboTwin. The LIBERO composition arms use $W = 130$ and 1,152 slots, matching the recent-2 FP arm at 0.94. Table 5 summarizes these configurations. The lag experiment applies a read-time filter selecting the single previous chunk at lag L and compares each arm with its own lag-1 reference. Other controls explicitly evict history and use the eviction-based recent-1 reference. The policy retains its current-chunk KV in either case. These implementations have separate references and must not be pooled as if they were identical experimental arms.

The original stale-chunk and content interventions use a separate same-node batch with three seeds per task ($n = 30$). The original lag-1, lag-2, recent-1, and two primary 64-token arms were extended to ten seeds per task ($n = 100$). Token pruning retains k of the 144 slots of the preceding chunk in every transformer block. Content interventions preserve slots, identifiers, token count, and rotary positions while replacing the specified K/V tensors. These controls distinguish content from slot occupancy; they do not establish that substituted states are in the training distribution.

RoboTwin replication. The controlled ladder uses `click_bell`, `stack_blocks_two`, and `place_dual_shoes`, with 34 planned seeds per task and 100 committed rollouts per arm after excluding two physically unstable seeds of the last task across every configuration. The driver stops before the KV write once an episode has ended. Recent-0 rollouts interrupted by GPU contention were repeated on a free device. The two tasks used in the offline memory study are different from these replication tasks.

Equivalence and bounded-read checks. We check action agreement, retention semantics, and read bounds separately:

1. *Six-chunk prefix.* The native `full` path uses $W = 30$ and 2,160 slots without a read filter. The `sparsekv` path uses $W = 130$, 9,360 slots, and `keep_lags=0..14`. On the same device, task 0 and seed 5 produce bitwise-identical action sequences for chunks 0–5 (`NEQ=6`). This seed is separate from evaluation seeds 10–19. The windows and pools differ, and six chunks reach neither configuration's capacity or retention limit. This comparison covers the pre-eviction prefix; native $W = 30$ first evicts around chunk 15.
2. *Native sparse-read reproduction.* The controlled sparse-read path exceeds the 10^{-2} maximum-absolute action-error threshold against native $W = 8$. A cache-state census identifies partial historical chunks under native slot-level eviction, whereas the read filter selects whole chunks (Table 6). The action sequences agree through chunks 0–3 and diverge when eviction begins. The before-call census records the pool states associated with these allocation differences.

3. *Bounded reads.* The evaluated implementation passes, with at most six chunks read against a bound of seven.

Full-trace retention replay. A separate replay checks token identity and position, represented by `(chunk, modality, ordinal)`, at the return of `update_cache`. Both arms use task 0, seed 10, $W = 130$, and 1,152 slots, with the retention cap set to one or two chunks. The comparison covers every recorded denoising forward over actual chunk IDs 0–17 and 0–18, respectively (Table 4). Each retained block has 128 video and 16 action tokens. Token identities and positions match the declared retention rule throughout each trace, including 1,224/1,224 forwards after eviction in each arm. The first eviction occurs at forward 73 (chunk 1) for recent-1 and forward 145 (chunk 2) for recent-2; forward indices here start at one.

Table 4: Full-trace cache-interface replay on task 0, seed 10. Both arms use $W = 130$ and 1,152 slots. Matches compare token identity and position with the declared retention rule. Prefill calls (`update_cache==2`) are excluded. Coverage is reported by actual chunk IDs.

Arm	Chunk IDs	All matches	After eviction	Excluded prefill	First eviction forward (chunk)
Recent-1	0–17	1,296/1,296	1,224/1,224	34	73 (1)
Recent-2	0–18	1,368/1,368	1,224/1,224	36	145 (2)

The two replays contain 72 denoising forwards per chunk and exclude 34 and 36 prefill calls, respectively. They validate token identity and position at the cache interface over the recorded denoising sequence.

The replay and the before-call census are separate runs with the same recent-1 store configuration. Replay compares the tokens returned by the cache update; the census captures pool masks before the wrapped call, including intermediate states between a commit and the next write. This capture-point distinction is retained in the census table.

Lag-filter read tracing. Separate lag-2 and lag-4 traces each report 540/540 forward passes with the read set $\{t, t - L\}$: the current chunk and the chunk at lag L. Both use task 0, seed 10, $W = 130$, and 9,360 slots; each episode contains 14 chunks and has success 0. Every recorded pass has `cmax=13`: the circular trace buffer covers the last chunk of one rollout per arm. During warm-up ($t < L$), the code's lag-L condition has no match and only lag 0 remains, making the first L chunks equivalent to recent-0. This warm-up rule is read from the code.

The matched-warm-up follow-up uses the same read rule in every arm while `cmax<8`: read lags $\{0, 1\}$, the current chunk and its immediate predecessor when available. Chunks 0–7 therefore use the common rule; at chunk 8, each arm switches to lags $\{0, L\}$. The warm-up length is eight because eight is the largest tested lag. All four arms complete 30 episodes, paired on the same task–seed units, with 28, 20, 4, and 0 successes at lags 1, 2, 4, and 8 (Table 9).

Two lag-1 units (task 0, seeds 10 and 11) interrupted by external GPU contention were rerun and included in the 30-unit paired sample.

The 16-bit KV hook is an identity and reproduces the unhooked action sequence bitwise in the reported check. Experiments use NVIDIA A6000/L40S nodes; each KV-precision comparison uses a same-node FP reference because bf16 rollouts can differ across GPU types.

Weight and activation realizations. The W4A8 configuration applies group-wise symmetric weights ($g = 128$, approximately 4.5 effective bits including group overhead) and per-token INT8 activations to 300 attention/FFN GEMMs, retaining 535 other tensors in FP32. It uses a simulated/dequantized path and $n = 500$ paired LIBERO-Long rollouts. The uniform weight sweep has 94–98 committed rollouts per precision. The activation sweep quantizes the same GEMM inputs by symmetric absmax, using either a scale per token or per 64 input channels per token; it uses seeds 10–19 on all ten tasks, paired with the matching FP and W4A8 units. The bit-allocation study compares action-sensitivity and random allocations with the same mean-bit budget and bit histogram ($n = 200$ paired). Both studies implement these quantization families on the same backbone; their configurations can differ from those in the cited work.

Table 5: Pool and retention settings. Pools are shared with prediction writes. The native LIBERO rows give upper bounds on complete historical chunks after the 128-slot prediction write. Native slot-level eviction can leave partial chunks; the $W = 8$ census and $W = 2$ token trace are in Appendix B. L denotes LIBERO-Long and R denotes RoboTwin.

Experiment	Suite	W	Pool slots	Retention / capacity
Native sweep	L	30	2,160	At most 14 complete chunks
	L	8	576	At most 3 complete chunks
	L	2	144	0 complete chunks; see Table 7
Native $W = 2$	R	2	272	One-chunk pool, shared with prediction write
Controlled recent-k	L	130	1,152	$k = 0, 1, 2$
	L	130	1,728	$k = 8$
Full, uncapped	L	130	9,360	No eviction over the evaluated episodes
Window sweep, recent-2	L	varied	1,152	$W \in \{2, 8, 30, 130\}$
Staleness/content	L	130	1,152	Intervention-specific read/retention rule
Controlled recent-k	R	130	2,176	$k = 0, 1, 2, 8$; recent-8 capacity-limited
Full, uncapped	R	130	17,680	No eviction over the evaluated episodes
KV precision	L	30	2,160	Native pool-limited history
KV precision	R	130	17,680	Uncapped native pool
Retention × precision	L	130	1,152	Recent-1 or recent-2

FastWAM. The second backbone has a 5B video expert, a 1B action expert, T5-encoded language, proprioceptive context, and three 384-pixel camera views. Q4_K replaces all 600 expert GEMM weight tensors with approximately 4.5-bit group-wise packed weights, dequantized for computation. RoboTwin evaluations use the released harness, seed 42, and 50 tasks × 100 episodes under clean and randomized protocols. The clean FP16 reference is an earlier independent run on different hardware; the randomized comparison uses a seed-paired FP16 run. Identical-seed FP16 runs differed by approximately 0.5–0.8 percentage points. The model-independent scripted expert governs seed acceptance. Five of six tasks in the smaller paired sweep used identical accepted seed sequences; one diverged after seven seeds through simulator nondeterminism.

LIBERO evaluations use four suites × ten tasks × 50 trials, paired. The SVDQuant realization uses a rank-32 low-rank branch, smoothing with $\alpha = 0.5$, two alternating iterations, and 4-bit weights/activations with activation groups of 64 on both experts. Calibration uses nine frames or 40 full trajectories (595 frames). Instrumented representative layers had at most 16 distinct levels per group under W4A4; the W4A16 check had 64 distinct values. Success was 0.965 or 0.968 against FP16 0.970. The 0.737 W4A4 baseline reported by Zhou et al. (2026) could not be reconciled with the available configuration information; the study treats these as different realizations, without a speed comparison to their Blackwell kernel result.

B Additional cache and precision results

B.1 Pool trace and per-task history results

Equation 1 gives $N_{\text{pool}} = 144$ at native $W = 2$. The 128-slot denoising write leaves only 16 slots available for history. Physical capacity alone does not determine which tokens the attention mask admits. At native $W = 30$, the token-level trace admits 1,792 committed video tokens and 224 committed action tokens, exactly 14 complete chunks. At $W = 8$, three complete chunks correspond to 384 video and 48 action tokens; the before-call census also records partial chunks during slot-level eviction (Table 6).

Native window-8 cache-state census. On task 0, seed 10, the native $W = 8$ census retains 364 records over chunks 12–17, all after the first capacity eviction at chunk 4. The circular buffer retains only this tail. Its four joint states are exhaustive (Table 6); video and action counts in that table refer to entries assigned to earlier chunks at the capture point. States with current-chunk entries are marked separately.

Table 6: Before-call cache census, distinct from the post-update retention replay in Table 4. All runs use task 0 and seed 10. The upper panel gives native $W = 8$ snapshot states; the lower panel lists the retained windows of the compared census runs. Snapshots precede the wrapped call and do not by themselves determine which entries a query subsequently reads.

State	Earlier chunk IDs	Earlier video	Earlier action	Current chunk present	Count
A	3	384	48	No	259
A+	3	384	48	Yes	5
H	3	256	48	Yes	95
X	4	384	64	No	5
				Total	364

Census configuration	W	Pool slots	Chunk IDs	Records
Native	8	576	12–17	364
Read filter, lags $\{0, 1, 2, 3\}$	130	9,360	12–16	294
Read filter, lags $\{0, 1\}$	130	9,360	12–17	364
Recent-1 cap	130	1,152	12–17	364

Native video and action writes use separate cache entries, with IDs $2c$ and $2c + 1$ for chunk c. Allocation evicts entries in ID order until the required number of slots is free. A 128-slot video write can therefore remove the oldest chunk's video while leaving its 16 action tokens, producing state H. In state X, the surviving extra action tokens belong to the historical chunk at lag four. The five X snapshots occur at recorded indices 36, 108, 180, 252, and 324, immediately after the current chunk index increments, with the same lag-four pattern each time.

States A and A+ account for 264/364 snapshots with three complete historical chunks; H and X account for 100/364 with a partial chunk. The partial states arise because native allocation can evict the video and action entries of one chunk separately. Whole-chunk filtering operates at a different granularity.

Window-2 denoising trace. The trace counts mask-admitted K/V tokens by modality in block-0 self-attention on task 0, seed 10, for chunks 0–3, using denoising forwards only (`update_cache=0`) (Table 7).

Across the native $W = 2$ trace, committed video is zero in 257/260 denoising forwards. The other three passes, numbered 51, 123, and 195, are the first reads after `_compute_kv_cache` commits the current chunk. They are spaced 72 calls apart, once per chunk, and read 128 current video tokens and 16 current action tokens before prediction slots exist; their previous-chunk set is empty. These passes remain in the denoising trace because `update_cache=0`. Historical video tokens are zero throughout all 260 forwards.

When the pool empties, `_next_cache_id` returns 0 when `mask.any()` is false, so numerical ID order cannot establish token age across resets. The co-residence statistic instead counts identifiers present simultaneously: a second identifier appears in 3/260 native forwards. Historical action reads sum to 48 across the trace. Controlled recent-1 reads a complete previous chunk—128 video and 16 action tokens—in 150/260 forwards, for cumulative historical counts of 19,200 video and 2,400 action token reads. These are sums over forwards, including repeated reads of cached tokens.

At fixed recent-2 retention, the $W = 2, 8, 30, 130$ sweep gives success 0.97 [0.93,1.00], 0.96 [0.92,0.99], 0.98 [0.95,1.00], and 0.94 [0.89,0.98], respectively ($n = 100$ per arm; 95% bootstrap CIs). At $W = 130$, recent-0/1/2/8/full give 0.00 [0.00,0.00], 0.99 [0.97,1.00], 0.94 [0.89,0.98], 0.97 [0.93,1.00], and 0.97 [0.93,1.00].

The native coupled ladder scored 0.975 at $W = 30$ (15-chunk pool), 0.965 at $W = 8$ (four-chunk pool), and 0 at $W = 2$ (one-chunk pool), with $n = 200$ per arm. The native labels therefore describe a window parameter, not the number of historical chunks that remain visible. Table 9 separates matched-warm-up results from the original lag ladder and the content controls. In the original, unmatched-warm-up ladder, lag-4 and lag-8 use the matched 30-unit subset of lag-1, whose success

Table 7: Exhaustive token-count states in block-0 self-attention on task 0, seed 10, chunks 0–3, for denoising forwards (`update_cache=0`). Token counts include only K/V entries admitted by the attention mask. The upper panel enumerates every observed joint state; the lower panel reports simultaneous chunk-ID co-residence and cumulative historical token-read counts across the same forwards. Committed tokens can belong to the current chunk, as explained below.

		Committed tokens			
Configuration	Pool slots	Video	Action	Prediction	Forwards
Native $W = 2$	144	0	0	128	197/260
		0	16	0	57/260
		0	16	128	3/260
		128	16	0	3/260
				Native total	**260/260**
Controlled recent-0	1,152	0	0	128	**200/200**
Controlled recent-1	1,152	128	16	128	150/260
		128	16	0	60/260
		0	0	128	50/260
				Recent-1 total	**260/260**

		Cumulative historical token reads	
Configuration	Second chunk ID present	Video	Action
Native $W = 2$	3/260	0	48
Controlled recent-0	0/200	0	0
Controlled recent-1	150/260	19,200	2,400

Table 8: LIBERO-Long per-task success for the controlled ladders; each cell contains ten rollouts. $W = 130$ unless specified.

Configuration	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9
Recent-0	0	0	0	0	0	0	0	0	0	0
Recent-1	1	1	1	1	1	1	1	1	.9	1
Recent-2	1	1	1	1	.9	1	.9	1	.7	.9
Recent-8	1	1	.9	1	.9	1	1	1	.9	1
Full	1	1	1	.9	.9	1	.9	1	1	1
$W = 2$, recent-2	1	1	.9	1	.9	1	.9	1	1	1
$W = 8$, recent-2	1	1	.9	1	.9	1	.9	1	.9	1
$W = 30$, recent-2	1	1	1	1	.9	1	.9	1	1	1

was 0.97; lag-2 uses the 100-unit reference of 0.96. These original references are not pooled with the new lag-1 arm.

Table 9: LIBERO-Long recency/content controls at $W = 130$, separating matched and unmatched warm-up. Intervals are 95% bootstrap CIs; m is median executed chunks. Every matched-warm-up arm completes 30 episodes; paired differences use all 30 common task–seed units. W/L/T denotes wins, losses, and ties against the matched lag-1 arm (Appendix A).

Arm	n	Success [CI]	m	Paired difference [CI]
Matched warm-up; reference is the new same-batch lag 1				
Lag 1	30	.933 [.833,1.000]	16	—
Lag 2	30	.667 [.500,.833]	27	$-.267$ [$-.467, -.100$]
Lag 4	30	.133 [.033,.267]	62	$-.800$ [$-.933, -.667$]
Lag 8	30	.000 [.000,.000]	62	$-.933$ [$-1.000, -.833$]
Original unmatched warm-up; original batch-specific lag-1 references				
Lag 1	100	.96 [.92,.99]	16	—
Lag 2	100	.36 [.27,.46]	62	$-.60$ [$-.70, -.50$]
Lag 4	30	.00 [.00,.00]	62	$-.97$ [$-1.00, -.90$]
Lag 8	30	.00 [.00,.00]	62	$-.97$ [$-1.00, -.90$]
Eviction controller; reference is recent-1 in the same 30-unit batch				
Recent-1	30	1.00 [1.00,1.00]	15	—
Chunk 0 only	30	.00 [.00,.00]	62	-1.00 [$-1.00, -1.00$]
Recent-0	30	.00 [.00,.00]	62	-1.00 [$-1.00, -1.00$]
K,V: channel mean	30	.00 [.00,.00]	62	-1.00 [$-1.00, -1.00$]
K,V: matched Gaussian	30	.00 [.00,.00]	62	-1.00 [$-1.00, -1.00$]
V: channel mean; K intact	30	.00 [.00,.00]	62	-1.00 [$-1.00, -1.00$]

Matched-warm-up contrast	W/L/T	Paired n	Paired difference
Lag 2 vs. lag 1	1/9/20	30	-0.267
Lag 4 vs. lag 1	0/24/6	30	-0.800
Lag 8 vs. lag 1	0/28/2	30	-0.933

Table 10: Same-unit comparison of unmatched and matched warm-up, paired on 30 task–seed units per contrast. These subset summaries are separate from the original 100-unit lag ladder. The comparisons are descriptive and do not decompose the original lag effect.

Lag	Unmatched warm-up SR	Matched warm-up SR	Paired change [95% CI]
2	0.30	0.67	$+0.37$ [$+0.13, +0.60$]
4	0.00	0.13	$+0.13$ [$+0.03, +0.27$]
8	0.00	0.00	$+0.00$ [$0, 0$]

For lag-2 and lag-4, matching warm-up improves success on the same task–seed units, with both paired intervals above zero. The lag-versus-lag-1 intervals after matching warm-up remain below zero. Thus early history availability contributes to the original ladder, while a staleness effect persists under the common warm-up rule. The original 100-unit lag-2 SR of 0.36 and the 30-unit subset SR of 0.30 describe different samples and are reported separately.

B.2 KV PRECISION, PRUNING, AND ALLOCATION

Each persistent chunk is fake-quantized once immediately after its KV write. K uses asymmetric min–max statistics per channel over 16-token groups; V uses per-token groups of 32 channels. Each group has one 16-bit scale and one 16-bit zero point. Prediction slots are never quantized. Storage remains bf16 (Section 2). For 30 blocks and 3,072 channels, LIBERO history payloads are 50.6/12.7/6.3 MiB per chunk at 16/4/2 bits, before scales and zero points; corresponding RoboTwin values are 95.6/23.9/12.0 MiB.

On LIBERO, K4/V4 gives 1/2/97 wins/losses/ties and K4/V2 gives 0/1/99 against FP. The K2/V2 comparison is sensitive to reference hardware: against an earlier FP arm on another GPU type (0.98),

Table 11: RoboTwin cache-capacity replication ($n = 100$ per arm). Controlled arms use $W = 130$. Recent-k has 2,176 slots, so recent-8 cannot hold eight complete previous chunks alongside prediction writes. Full uses 17,680 slots with no explicit retention cap; native $W = 2$ has 272 slots. Paired difference CIs are reported for the three positive-retention arms.

Configuration	Success [CI]	Difference vs. full	Difference CI
Native $W = 2$	.05 [.01,.10]	$-.89$	—
Recent-0	.00 [.00,.00]	$-.94$	—
Recent-1	.98 [.95,1.00]	$+.04$	$[-.01, .10]$
Recent-2	1.00 [1.00,1.00]	$+.06$	[.02,.11]
Recent-8	.99 [.97,1.00]	$+.05$	[.01,.09]
Full	.94 [.89,.98]	—	—

Table 12: Complete KV-precision comparisons, $n = 100$ per arm, each against the same-node FP-KV reference: native $W = 30$ on LIBERO, $W = 130$ on RoboTwin. Cosines apply to LIBERO K/V.

Precision	Success [CI]	Difference vs. FP [CI]	K/V cosine
LIBERO-Long			
FP (16/16)	.99 [.97,1.00]	—	1.000/1.000
K4/V4	.98 [.95,1.00]	$-.01$ $[-.05, .02]$	.999/.997
K4/V2	.98 [.95,1.00]	$-.01$ $[-.03, .00]$	.999/.932
K2/V2	.93 [.88,.98]	$-.06$ $[-.12, -.01]$	.979/.932
RoboTwin			
FP (16/16)	.99 [.97,1.00]	—	—
K4/V4	.99 [.97,1.00]	$+.00$ $[-.03, .03]$	—
K4/V2	.96 [.92,.99]	$-.03$ $[-.07, .00]$	—
K2/V2	.96 [.92,.99]	$-.03$ $[-.08, .01]$	—

its difference is $-.05$ with CI $[-.11, .01]$, whereas the matched same-node comparison excludes zero. The direction is consistent, but significance is not portable between these references.

The retention–precision composition in Section 4 includes group overhead. At the fine K/V groups of 16/32, scales and zero points add 50% of the K4/V2 payload; two chunks occupy approximately 28.5 MiB in storage-equivalent accounting. Coarser groups (48/64 and 144/128) reduce this to 22.7 and 20.5 MiB, respectively (scale overhead 19% and 8%), but score 0.940 [0.890, 0.980] and 0.950 [0.900, 0.990]. Against native $W = 30$ FP, their paired differences are -0.050 $[-0.100, -0.010]$ and -0.040 $[-0.080, -0.010]$, with 0/5/95 and 0/4/96 wins/losses/ties. Both intervals exclude zero. Each comparison uses 100 matched task–seed units and 10,000 bootstrap resamples. Table 13 lists these settings alongside the single-chunk settings, all against that same native FP reference. The operating point follows the descriptive selection rule in Section 4.

Table 13: History-only storage equivalents and operating-point selection. Ratios use the native 714.4 MiB history-slot reference; displayed storage and ratios are rounded independently. g gives K/V group sizes. All paired differences use native $W = 30$ FP (N). Brackets give 95% CIs.

Configuration	MiB	Ratio	SR	Paired difference	Ref.
Recent-2 K4/V2, $g = 16/32$	**28.5**	$25.1\times$	**0.99**	$+0.00$ $[-0.03, +0.03]$	N
Recent-2 K4/V2, $g = 48/64$	22.7	$31.5\times$	0.94	-0.05 $[-0.10, -0.01]$	N
Recent-2 K4/V2, $g = 144/128$	20.5	$34.9\times$	0.95	-0.04 $[-0.08, -0.01]$	N
Recent-1 K4/V2	14.2	$50.2\times$	0.96	-0.03 $[-0.07, +0.00]$	N
Recent-1 K2/V2	11.1	$64.5\times$	0.93	-0.06 $[-0.12, -0.01]$	N

At 64 tokens, blind and contiguous selection tie on 82/100 paired units, with mean difference 0 and CI $[-.08, .08]$. The degradation is evidence about these hand-designed selectors, not an impossibility result for learned token compression. Adding a first-chunk attention sink to recent-1 scores 0.94 [.89,.98], $-.05$ versus full FP (CI $[-.10, .00]$) and $-.05$ versus recent-1 (CI $[-.10, -.01]$, 0/5/95 wins/losses/ties). A heavy-hitter variant retains recent-1 plus the chunk with the greatest accumulated attention mass, scoring 0.97 [.93,1.00]: $-.02$ versus recent-1 (CI $[-.06, .02]$) and $+.03$ versus the

Table 14: Complete token-pruning results in the retained LIBERO chunk. The $n = 30$ unpruned reference is 1.00; the matched $n = 100$ reference for the two extended 64-token arms is 0.99. m is median executed chunks.

Selection	n	Success [CI]	m	Difference [CI]
144: unpruned	30	1.00 [1.00,1.00]	15	—
64: blind stride	100	.89 [.82,.95]	15	$-.10$ [$-.16, -.05$]
64: 16 action + 48 contiguous	100	.89 [.83,.95]	17	$-.10$ [$-.16, -.05$]
64: 16 action + 48 strided	30	.80 [.63,.93]	19	$-.20$ [$-.37, -.07$]
32: blind stride	30	.57 [.40,.73]	25	$-.43$ [$-.60, -.27$]
16: action only	30	.00 [.00,.00]	62	-1.00 [$-1.00, -1.00$]
16: latent only	30	.00 [.00,.00]	62	-1.00 [$-1.00, -1.00$]
8: blind stride	30	.00 [.00,.00]	62	-1.00 [$-1.00, -1.00$]

sink arm (CI $[-.02, .08]$). These trials did not demonstrate an advantage for either additional landmark rule.

For weights, uniform W4/W3/W2 give 0.98/0.60/0.00, with the W3 CI [.50,.69]. At a matched 3.5-bit budget, action-aware allocation scores .90 versus .84 for random allocation ($+.06$, CI $[-.005, .125]$), with the gain concentrated in one task. A 3.0-bit 4/2-bit tensor allocation scores zero. For activations with W4 fixed, A6/A5/per-token A4/group-64 A4 score .95/.72/.00/.92. Their reported differences versus the matching FP reference are $-.01$ $[-.06, .04]$, $-.24$ $[-.33, -.15]$, and $-.04$ $[-.10, .02]$ for A6, A5, and group-64 A4, respectively. W16A4 also scores zero, separating the per-token A4 failure from weight quantization. W4A8 on the larger paired run gives .980 versus .972, difference CI $[-.012, .028]$ and success-ratio retention 1.008. These budgets and sample sizes should remain attached to their own references.

C AUXILIARY-MEMORY DESIGN

QSDM is an auxiliary output-residual memory used to diagnose teacher-action prediction. At each memory event, the production three-camera VAE feature pipeline produces a 96-dimensional pooled visual feature, transformed by frozen TRAIN z-score and PCA constants to $x_t \in \mathbb{R}^{64}$. Proprioception is excluded because it can trivially derive the target. We verified the pre-PCA feature against the production latent on 9,270 TRAIN+DEV anchors (maximum absolute difference zero). Recent local context is $L_t = [x_{t-1}, x_t]$ at A8.

A shared projection $z_t = Bx_t$, $B \in \mathbb{R}^{8\times 64}$, writes eight values into a 64-slot circular ring. Each slot is quantized once at write time to four bits per value with an FP16 scale. At time t, the read-before-write schedule retrieves fixed lags $D = \{4, 8, 16, 32, 64\}$; invalid warm-up lags contribute zero. The delayed branch is

$$r_t = \sum_{d\in D} C_d \,\mathrm{dequant}(z_{t-d}), \qquad C_d \in \mathbb{R}^{16\times 8}.$$

It joins a recent-2 local readout. With teacher and local student frozen, training minimizes MSE on $r_t^\star = \mathrm{sg}(a_t^T - a_t^L)$. In policy integration, the residual modifies only the first action step's 16 active dimensions; the rest of the chunk is unchanged. The canonical design has no recurrence, error feedback, attention, event gate, or learned lag selection. Later gate experiments are explicitly post hoc.

The ring has fixed capacity independent of episode length, with $\min(t, 64)$ logically valid slots. Its 256-byte A4 payload and 128-byte FP16 scales total 384 bytes. Adding 128 bytes of recent-2 A8 context and 17 logical metadata bytes gives the reported **529-byte logical temporal state**. The packed implementation instead uses an eight-byte ring header, so ring plus recent context occupies **520 physical bytes**. These are different accounting conventions for the same auxiliary design; neither includes the policy's KV cache or model weights. The trainable readout contains $512+2048+640+16 = 3216$ parameters (projection, local head, delayed heads, bias), and the frozen feature adapter has 6,432 constants. The ring is rebuilt at runtime rather than stored in checkpoints.

Writing each slot once prevents repeated quantization of that stored vector as its age increases. It does not make four-bit storage mathematically lossless. The experiments below show negligible

observed fidelity changes for several ring models, in contrast to larger changes when recurrent state is requantized every step. Exact sparse retrieval has a different requirement: without reload or recomputation, an exact moving-lag read must preserve every still-needed state until its final read. Thus storage scales with the largest required delay (15 previous chunks for our indexing of $\{4, 8, 16\}$), even if each query reads only a few lags. Lossy projected rings are not exempt from counting their full allocated capacity.

D OFFLINE COMPARISONS AND FULL-MODEL INTEGRATION

D.1 PROTOCOL AND COMPLETE REPRESENTATION COMPARISON

The offline target is the real LingBot-VA teacher's first-step action at attention window 72, normalized using TRAIN statistics over 16 active dimensions. The two tasks are `battery_try` and `observe_and_pickup`. Eighty episodes form TRAIN and ten episode-disjoint episodes form DEV, from the same tasks. Long-context RMS includes anchors beyond lag 32. Oracles and fixed bases use ridge readouts; QSDM uses deterministic single-thread AdamW. Recovery is

$$\frac{\mathrm{RMS}(H1) - \mathrm{RMS}(\mathrm{method})}{\mathrm{RMS}(H1) - \mathrm{RMS}(H3)}.$$

H1 is recent-2 and H3 is an uncompressed 512-dimensional explicit-lag ridge reference. H3 is not an accuracy ceiling; recovery can exceed one. These are in-task prediction metrics, not task success, and the target teacher did not solve these tasks in the limited qualification rollouts (Appendix F).

Table 15: Canonical offline comparison. RMS is normalized long-context DEV error. State sizes are as reported: ring-only rows exclude the shared local state and metadata, whereas the 529-byte canonical row includes them.

Method	RMS	Recovery	Temporal state	Fit parameters
Recent-2 (H1)	.206	.00	128 B (A8)	2.1K ridge
EMA	.197	.10	256-d full-rank	4.1K ridge
Complex-pole basis	.174	.37	2048-d full-rank	32.8K ridge
HiPPO-LegT	.167	.46	2048-d full-rank	32.8K ridge
Recent-8 (H2)	.154	.63	512 B	8.2K ridge
QSDM W4/A4, canonical	.144	.72	529 B logical	3.2K AdamW
QSDM FP32	.143	.74	FP state	3.2K AdamW
QSDM, 1500 steps	.131	.87	384 B ring	3.2K AdamW
H3 explicit-lag reference	.121	1.00	512-d uncompressed	8.2K ridge

The canonical QSDM result was frozen before the later model-selection experiments. Fixed EMA, complex-pole, and HiPPO bases have larger state/readout budgets but lower recovery here. This comparison does not cover trained versions of all state-space architectures. Table 16 reports the stronger trained alternatives, including those that outperform the canonical sparse-delay configuration.

Trainable comparisons share the 2,064-parameter recent-2 local readout. Their memory branches have 1,296 parameters for the real diagonal SSM and complex LRU, 2,336/1,168 for selective SSMs of rank 16/8, 1,152 for the dense lag kernel, and 1,216 for ring attention. Hence the rank-16 selective model has a larger parameter budget, while the rank-8 version is approximately matched to QSDM. Training uses AdamW (learning rate 10^{-3}, weight decay 10^{-4}, full batch, 1,500 steps, three seeds), followed by a ridge readout refit with $\lambda = .1$ on TRAIN. The maximum cached episode length is 223 events.

Reported FP32/A4 RMS pairs are QSDM-400 .143/.143, QSDM-1500 .131/.132, dense kernel .120/.121, selective-$r16$ .121/.138, selective-$r8$ .143/.154, LRU .133/.180, real diagonal SSM .164/.176, and ring attention .179/.178. These reruns are distinct from the frozen .144 canonical evaluation in Table 15. Recurrent models requantize state at each step; ring models quantize slots at write time. Quantization effects depend on the architecture and are not uniformly .02–.05. Episode-paired mean-error differences for median seeds versus QSDM-1500 are $-.045$ $[-.077, -.014]$ for the dense kernel, $-.047$ $[-.094, -.003]$ for selective-$r16$, $-.048$ $[-.102, .000]$ for LRU, and $-.013$ $[-.064, .029]$ for selective-$r8$. These episode-level comparisons are not differences between the

Table 16: Post-hoc trainable memories and gates. RMS is the reported mean [minimum–maximum] over three seeds, except the two gate controls. Trainable-memory rows use FP32 RMS; their A4 results are stated in the text. Gate rows use A4. The H3 reference is .121 (recovery 1.00).

Method	RMS	Recovery	State	Fit params
Dense lag kernel	.120 [.111–.126]	1.01	384 B ring	3.2K
Selective SSM, $r = 16$	.121 [.108–.130]	1.00	64 B FP32	4.4K
Selective SSM, $r = 8$	.143 [.129–.164]	.74	32 B FP32	3.2K
Learned LRU	.133 [.126–.139]	.86	64 B FP32	3.4K
Real diagonal SSM	.164 [.161–.166]	.50	64 B FP32	3.4K
Ring attention	.179 [.169–.187]	.31	384 B ring	3.3K
QSDM + rank-4 input gate	.100 [.096–.104]	1.25	384 B ring	3.6K
QSDM + input gate, 4000 steps	.086 [.080–.092]	1.40	384 B ring	5.8K
QSDM + gate + per-lag B	.083 [.082–.085]	1.44	384 B ring	7.8K
QSDM + read/write gates	.070 [.065–.076]	1.59	384 B ring	6.3K
Gate on recent-2 only	.164	.48	—	12.3K
Gate on dense recent-8	.113	1.08	512 B	7.7K

pooled RMS numbers in the table. With ten DEV episodes and three seeds, they remain limited evidence.

D.2 Ablations, gates, and robustness

Table 17: Canonical-trainer ablations. Dashes indicate configurations for which only recovery was recorded. Reported recovery/byte values are retained rather than recomputed from rounded cross-experiment references.

Variant	A4 RMS	Recovery	Temporal bytes
Canonical, trained rank 8	.1442	.722	529
Random frozen B, five seeds	—	.346 (.29–.41)	529
Random trained B, five seeds	—	.797 (.73–.88)	529
Near + far lags, rank 8	.1340	.841	533
Dense lags 1–8, rank 8	.1540	.607	189
Dense lags 1–64, rank 1	.1527	.622	396
Dense lags 1–64, rank 2	.1470	.689	396
Canonical sparse lags, rank 4	.1484	.672	401

Learning the projection matters; the frozen random projection is weaker, while trained random initializations improve the mean result over the frozen canonical SVD initialization. Near-lag reads improve this primitive metric, but not the full-policy residual below. The dense kernel comparisons show that the sparse lag set is not uniquely responsible for the prediction gain.

Post-hoc gate selection uses an inner 64/16 split of TRAIN and evaluates each configuration once on DEV. Canonical A4 RMS at 400/1500/4000 steps is .143/.132/.123. Other reported variants reach .123 with near+far lags, .121 with per-lag projections, .111 for a dense kernel at 4000 steps, .108 for rank 16 (640-byte ring), and .138 for rank 4 (256-byte ring). Group-L1 pruning selects lags $\{11, 12, 40, 41, 48\}$ and gives .127. A sigmoid input gate on the 40-dimensional delayed read gives .090 at 1500 steps and .086 at 4000; combining read and write gates gives .070. A write gate alone gives .085, while 12,000 steps worsen the combined result to .092. The controls in Table 16 indicate that nonlinearity on recent context alone does not reproduce the delayed-memory gain.

We checked reproducibility, data sensitivity, and quantization robustness for the fixed QSDM configuration. Five independent training processes reproduced identical parameters and loss traces, with RMS .1442 and recovery .722. Training on 75% subsets gave median/worst recovery .782/.715. QSDM improved over recent-2 on 9/10 DEV episodes, with a bootstrap gain of 29.2% [16.7,36.3]%. The top-three lag rankings agreed between TRAIN and DEV (3/3 overlap; Spearman correlation .70).

Perturbing each lag by ± 1 gave median/worst recovery .701/.663; removing a single lag increased RMS by .004–.012, with the largest change at lag 32. A4 retained at least .925 of the gain across six

runs. These checks used 529 bytes of logical state, quantized each stored slot once, and excluded proprioception and DEV fitting. Projection transfer was asymmetric: training on battery and testing on observe retained .013 of the gain, versus .616 in the reverse direction.

The gate cross-validation uses nine task-stratified folds over 90 cached episodes, with 5+5 held-out episodes per fold. Its PCA is the frozen canonical basis fit on the original 80 TRAIN episodes; it is not refit within each fold. Thus this tests refitting the memory conditional on fixed preprocessing, and is not a fully nested evaluation of the complete learned pipeline. Pooled held-out RMS is .203/.147/.120/.069 for recent-2/recent-8/QSDM/gated QSDM. The gated model ranks first in 9/9 folds and wins on 87/90 episodes; the reported episode-paired difference is $-.087$ $[-.102, -.072]$. Mean episode RMS changes .106→.059 on battery and .263→.137 on observe.

Additional gate checks report five-seed RMS .065–.077; 10/10 DEV episode wins with CI $[-.116, -.034]$; 75% TRAIN subsets .069–.096 versus .121–.128 for the canonical structure; lag-jitter RMS .071/.080 versus .130/.119; A3 ring results .067–.081 and A2 results .083–.099 (canonical A3/A2 .126/.129). Task transfer remains poor: battery→observe RMS is .62 gated versus .68 canonical, and the reverse .46 versus 1.20. These task-conditioned results do not establish a transferable memory basis.

D.3 Integration and closed-loop scope

The live adapter uses the same 80-episode training set structure, frozen teacher/student, and a residual head. Long-context teacher-relative RMS is .607 for recent-2 (reported history KV 220 MB), .554 with the frozen offline adapter, .459 with the live-trained adapter, and .422 for recent-8 (881 MB). The live-trained gain is 24.5% [18.3,30.6]%, positive on 9/10 DEV episodes and approximately 24–25% on both tasks. All-context RMS is .444 versus .569; recovery toward recent-8 is 80.3%. The adapter adds 520 physical bytes of auxiliary state. W4/A4 gain retention is 1.007. These MB figures belong to this separate integration configuration and should not replace the LIBERO history accounting in Section 4.

Retraining with lags $\{3, 4, 5, 6, 7, 8, 16, 32, 64\}$ gives .4682 versus the reproduced canonical .4585, with gain 22.9% and W4/A4 retention 1.011. Gated live adapters give .456–.457 versus .458 for the linear head, with all variants initially selecting the 100-step checkpoint. Under the live protocol's 68/12 TRAIN split and early stopping, offline linear and read/write-gated models still separate at their selected later checkpoints (.123 versus .069–.078), although both are similar at step 100. Lowering the live learning rate tenfold and extending patience yields .452 for the linear head at step 500 and .454–.455 for the gated head near step 900; gate-only training on a frozen canonical adapter gives .463. These checks did not recover the large offline gate advantage on the live residual. Internal cross-chunk attention is a plausible structural explanation for the remaining recent-8 advantage, not a mechanism established by these output-head ablations.

On the separate LIBERO closed-loop evaluation, the frozen residual is fused into the deconfounded recent-1/recent-2 policies at $W = 130$, using the same 100 paired task–seed units as the corresponding references. Recent-1 changes .99→.97 (difference $-.02$, CI $[-.06, .02]$, 1/3/96 wins/losses/ties); recent-2 changes .94→.98 ($+.04$, CI $[-.01, .10]$, 6/2/92). The latter gain concentrates in task 8 (.7→1.0). Neither comparison resolves a significant gain, and neither supplies a zero upper bound. The offline tasks, target, and metric differ from these closed-loop trials. Near-lag and gate integration tests above measure prediction error, and are not additional closed-loop success experiments.

E Runtime accounting and external code audit

E.1 Packed-weight runtime measurement

The measured C++/ggml implementation keeps LingBot-VA weights on one L40S GPU and runs the Python VAE/T5/simulator driver on a second GPU of the same node. Q4_K and F32 runs use LIBERO-Long task 0, seed 0, eight chunks each, and execute back to back on an otherwise unshared GPU pair. Per-process VRAM is sampled every two seconds and attributed by GPU UUID. A chunk makes two forward requests (prefill/KV and denoise); latency is measured at the shared-memory request/response bridge.

Table 18: Controlled packed-weight measurement with floating-point computation.

Quantity	Q4_K	F32
Resident weight store (GB)	2.76	19.63
Serve-process peak VRAM (MiB)	3,682	19,510
Checkpoint size (GB)	3.17	18.96
Median chunk time, chunks 1–7 (s)	447	448
Chunk 0, including initialization (s)	713	377

Post-warm-up chunk times are 356, 377, 394, 447, 501, 542, 568 seconds for Q4_K and 336, 374, 403, 448, 523, 569, 649 seconds for F32. The weight-store reduction is 7.1× against F32, or 3.6× against the approximately 9.8-GB bf16 weight store; the serve-process peak reduction is 5.3×. The unchanged driver has a reported peak of 22,446 MiB and median of 2,812 MiB. The memory ratio therefore applies to the serving process. The implementation is dominated by non-GEMM work and shows no measured median latency improvement.

E.2 External window/pool audit

We audited official released code on 2026-09-11 and rechecked it on 2026-09-13 without executing these systems. The signature is a common parameter setting both the attention radius and allocated KV capacity while the generated block is written into the same buffer. Three of the five external systems exhibit this signature in the inspected versions (Table 19). This is a code-structure observation, not evidence that their reported generation results are incorrect.

Table 19: Window/pool coupling in the audited implementations. Paths and expressions refer to the inspected versions, not an assertion about future releases.

System	Allocation/mask relationship and source location	Coupled
LingBot-VA	`total_tolen = (attn_window//2) * tokens_per_chunk`; `wan_va/modules/model.py:664`.	Yes
Self-Forcing	`kv_cache_size=local_attn_size*frame_seq_length`; `pipeline/causal_inference.py:285`; mask uses `local_attn_size*1560`.	Yes
LongLive	Same allocation form in `pipeline/causal_diffusion_inference.py:777`; mask uses `local_attn_size*880` in `wan_5b/modules/causal_model.py:317`.	Yes
Forcing-KV	`kv_cache_size=local_attn_cfg*frame_seq_length`; `pipeline/causal_inference_forcingkv.py:135`.	Yes
Rolling Forcing	Coupled lines commented out; fixed `kv_cache_size=1560*24`; `pipeline/rolling_forcing_inference.py:343--349`.	No
CausVid	Fixed `torch.zeros([B,32760,12,128])`; `causvid/models/wan/causal_inference.py:56`.	No

The inspected eviction arithmetic subtracts the new block size from the available history: visible history is nominal local attention size minus frames per generated block, subject to protected sink regions. In Self-Forcing the default one-frame block means a local size of one leaves no previous frame. The coupled video implementations' protected sinks and once-per-block writes differ from LingBot-VA's repeated prediction-slot allocation during denoising. These differences prevent direct extrapolation of the robotics failure magnitude. At the time of the audit (2026-09-11), we did not locate public HiMem-WAM code. A public repository is available as of 2026-09-24, but its allocation path is not included in this audit.[1]

Comparison taxonomy. ActQuant and QVLA motivate action-guided/adaptive bit allocation; QuantVLA motivates the W4A8 DiT family. Our W4A8 and bit-allocation studies implement these families on the same backbone. SVDQuant supplies low-rank/smoothing W4A4, realized

[1] https://github.com/Agentic-Intelligence-Lab/HiMem-WAM

on FastWAM here. QuantWAMs studies calibration granularity on the same model families with a different PTQ realization. VLA-Cache reuses visual computation to reduce latency, while VL-Cache compresses VLM KV. SmoothQuant and DuQuant are generic quantization references. These differing backbones and objectives preclude ranking them by cross-paper success rates.

F Reproducibility and uninformative preliminary tests

Determinism and source artifacts. The trainer uses single-thread OpenBLAS configured before NumPy import, deterministic PyTorch, and a canonical-sign SVD-initialized projection loaded from a frozen artifact. Five independent training processes produced identical parameters and loss traces. The ring simulator and vectorized evaluator agreed within 1.3×10^{-15} and reproduced the canonical .1442 RMS.

Feature preprocessing aligns PCA component signs to the frozen basis used to train B (maximum deviation 2×10^{-4}). Unaligned evaluations with varying TRAIN concatenation order produce .147, .151, and .172 RMS against the .1442 reference. We check trainer determinism and feature-pipeline reproducibility separately.

The numerical results and configurations are recorded in result manifests, which are intended for release with the code.

Native small-pool residual test. A LIBERO test added the frozen QSDM output residual to the native one-chunk-pool configuration. Both arms scored zero over 200 rollouts, while full and native recent-8 scored .975 and .965. This native pool cannot retain a complete previous chunk during denoising, so both arms are subject to the pool-capacity failure. Appendix D evaluates the residual under controlled recent-1/recent-2 retention.

Preliminary cross-suite test. A zero-shot RoboTwin test used 20-chunk rollouts and produced residual norms around 13 with no successes. The reference policy also failed, leaving this test without a competent control policy.

Teacher competence on offline tasks. Under the reported runtime, the teacher succeeded on 0/4 rollouts for each of `battery_try` and `observe_and_pickup`, and 0/16 across the four qualified RMBench tasks overall. The benchmark's scripted expert succeeded on 5/5 planned demonstrations. The offline target therefore measures fidelity to this teacher's action distribution; the qualification provides no evidence of competent task execution by that teacher.

Preliminary learned KV decoder. A global FP16 code with a factorized per-layer KV decoder plateaued at key/value reconstruction cosines .75/.48, with similar train and validation behavior, below its prespecified .95 fidelity threshold. No closed-loop evaluation of that decoder was measured. This is a capacity observation about the tested decoder family, not a closed-loop negative result or a bound on alternative latent-memory architectures.

G COMPLEMENTARY WEIGHT AND ACTIVATION RESULTS

Weight and activation precision concern a different part of the memory budget. Table 20 summarizes the broader feasibility of low-bit VLA representations. On LingBot-VA, a same-backbone W4A8 realization of the post-training quantization family gives 0.980 success versus 0.972 for FP over 500 paired rollouts (difference CI $[-0.012, +0.028]$). Uniform W3 and W2 degrade, and per-token A4 fails; switching A4 to scales over 64 input channels recovers 0.92. The detailed bit-allocation and activation sweeps assess the existing quantization families under these backbone-specific configurations.

Table 20: **Selected weight and activation results.** n is per arm. LingBot-VA uses simulated quantization. FastWAM uses packed weights, with genuine activation quantization in the SVDQuant rows but floating-point compute. Clean RoboTwin references are cross-run; randomized RoboTwin and LIBERO arms are seed-paired. These results do not replicate the history intervention on FastWAM.

Backbone / suite	Representation	FP $\rightarrow$ quantized SR	n
LingBot-VA / LIBERO-Long	W4A8 ($\sim$4.5-bit W)	0.972 $\rightarrow$ 0.980	500
FastWAM / RoboTwin-50, clean	Packed Q4_K	0.923 $\rightarrow$ 0.920	5,000
FastWAM / RoboTwin-50, randomized	Packed Q4_K	0.921 $\rightarrow$ 0.916	5,000
FastWAM / LIBERO-40	W4A4; 9-frame calibration	0.970 $\rightarrow$ 0.965	2,000
FastWAM / LIBERO-40	W4A4; 40-trajectory calibration	0.970 $\rightarrow$ 0.968	2,000

FastWAM extends this numerical evidence to a second backbone. Packed Q4_K weights remain close to the FP reference over 50 RoboTwin tasks, and SVDQuant W4A4 (Li et al., 2025) gives 0.965–0.968 versus 0.970 on 40 LIBERO tasks. These are our realizations, with calibration and compute-path details in the appendices; they do not imply reproduction of another paper's deployment kernel or throughput. In a separate LingBot-VA serving measurement, packed weights reduce the resident weight store from 19.63 to 2.76 GB and the serving process's peak memory from 19,510 to 3,682 MiB relative to F32. The driver runs separately, and median chunk latency is essentially unchanged (448 versus 447 s). Measurement details are in Appendix E.