TEST-TIME OPTIMIZATION IS NOT TEST-TIME MEMORIZATION: AUDITING NEURAL MEMORY ON EGOCENTRIC VIDEO

Anonymous authors
Paper under double-blind review

ABSTRACT

A test-time memory writes a stream into adaptive parameters by descending a self-supervised loss during the forward pass. This recipe can succeed as optimization while failing as memorization, and the training loss cannot tell the two apart. We audit a Titans-style MLP memory over frozen DINOv2 features. We first formalize when a recall objective cannot identify memory use, through a sufficient condition for a write-free global optimum and a Bayes-risk gap that states when causal history can help beyond the query. Trained for recall on i.i.d. image sequences, the memory approaches that optimum. The loss falls from 1.24 to 0.12 while the write rate contracts 200-fold, the final state loses input dependence, and retrieval drops below that of the same architecture at random initialization. Egocentric video adds a second shortcut. Frozen features remain correlated twenty seconds apart, and a causal convolution in the key exposes recent features to the query. Under a past–future windowed objective, the standard diagnostics look positive, and controls explain each of them away. At the default horizon, a copy of the oldest feature in the key window matches the far-past gain, a stateless filter with five coefficients per offset matches or beats the model at every offset, and the seen/unseen AUC ranges from 0.23 to 0.92 depending on which video is written. With a horizon beyond the key window, the model outperforms the filter twelve to seventeen seconds back while relying on its own state, yet its trace of individual items stays weak. The seen/unseen AUC detects neither case. We distill the audit into the controls that any storage claim about a test-time memory should report.

1 INTRODUCTION

A test-time memory replaces an ever-growing context with a fixed-size state that changes while a sequence is processed. In test-time training (TTT), that state can itself be a neural network: its weights are updated by a self-supervised objective and later queried as an associative memory (Sun et al., 2025; Behrouz et al., 2025b;a). This approach has shown promising results on long-context language modeling, and it is attractive for continuous video, where storing every frame is impractical and new queries may arrive long after an event. However, it also creates an attribution problem: one outer loss jointly trains the key and value projections, gates, initial memory, and readout. If any fixed-state path can predict the target, the loss may fall even when online writes contribute nothing.

Such an attribution problem is especially prominent for video, because consecutive frames usually share scene, objects, and viewpoint. Frozen representations therefore remain similar for tens of seconds. Copying the current feature is a strong temporal predictor, and when the key is a causal temporal convolution, it exposes several recent features directly to the query. A prediction loss can measure local dynamics rather than retained history. A low associative inner loss is not decisive either. It shows that the memory fits the current key–value pair, not that an earlier item remains recoverable after interference.

We treat a Titans-style neural memory as a case study. Rather than evaluating a downstream task such as video understanding, we ask a narrower question: *what evidence identifies the adaptive state as the carrier of past information?* We organize the answer into three levels of evidence. First, the target must contain information not already available from the query. Second, the learned

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.

state must depend on the streamed content. Third, changing or disabling that state must change content-matched retrieval.

Our contributions are:

- We formalize when a recall objective cannot identify memory use: a sufficient condition for a write-free optimum, and a causal-history risk gap that states when history can help beyond the query.
- We show an empirical collapse hidden by a falling recall loss: the write gate closes, final states lose input dependence, written and unwritten errors coincide, and training destroys retrieval present at random initialization.
- We organize a state audit for correlated video and apply it. Copy-relative gain, seen/unseen AUC, inner loss, gate magnitude, and past–future asymmetry each look informative alone. State swaps, receptive-field copies, a stateless filter, a rotation of the written video, and within-video negatives show that the model beats this filter only when the horizon reaches beyond the key window.

Together, these contributions recast objective design for test-time memory as an identifiability problem. An objective should be judged not only by the loss it attains, but also by whether success requires context that can reach the prediction only through the adaptive state. This leads to a minimum evaluation standard: query-only baselines, interventions on the online writes of the same checkpoint, content-matched negatives, and uncertainty over independent streams. The same audit extends, in principle, to fast-weight networks, recurrent latent states, and compressed caches, which all face one question: *what information was retained because the system adapted at test time?*

2 Related Work

Test-time memorization. Storing associations in weights predates deep learning. Correlation-matrix memories and Hopfield networks write key–value pairs with a fixed Hebbian rule, and their capacity before interference is well characterized (Kohonen, 1972; Hopfield, 1982; Amit et al., 1985). Fast-weight programmers and linear attention learn the keys and values they add to the state (Schmidhuber, 1992; Katharopoulos et al., 2020; Schlag et al., 2021), and the delta rule and its gated variants make the write corrective (Yang et al., 2024; 2025). TTT replaces these fixed rules with gradient descent on a self-supervised loss and interprets the recurrent state as fast weights optimized on the current context (Sun et al., 2025). Titans makes the state a deep associative memory with a surprise-driven update, momentum, and a forgetting gate, while ATLAS expands the update from a token to a context and introduces larger feature maps and a Muon-style optimizer (Behrouz et al., 2025b;a). MIRAS separates this design space into memory architecture, attentional-bias loss, retention rule, and online optimizer, and explores losses beyond squared error (Behrouz et al., 2026). LaCT uses very large chunks to improve hardware utilization and scale nonlinear fast weights, with applications to novel-view synthesis and video diffusion (Zhang et al., 2026b). TNT instead decouples large-chunk pretraining from small-chunk fine-tuning through global and resettable local memories (Li et al., 2026). These works primarily improve capacity, optimization, or throughput. Our concern is complementary: whether the training and evaluation objective identifies use of the adaptive state.

Evaluating memory use. A low loss does not show that a model uses its context. Long-range language models gain little from context beyond a few thousand tokens (Sun et al., 2021), and single frames often solve video tasks framed as temporal (Buch et al., 2022; Lei et al., 2023). Both are instances of shortcut learning (Geirhos et al., 2020). Associative-recall and needle-in-a-haystack suites therefore separate the key from the value (Arora et al., 2024; Hsieh et al., 2024; Kuratov et al., 2024). We bring this concern to the adaptive state itself, where the shortcut is the query.

Memory for visual streams. Memory-augmented networks read from and write to an external store through a learned addressing head (Graves et al., 2014; 2016), and long-context transformers cache or compress past segments (Dai et al., 2019; Bulatov et al., 2022; Munkhdalai et al., 2024). Video models mostly rely on fixed write rules, from multiscale feature caches (Wu et al., 2022) and sparse token memories for long videos (Song et al., 2024) to spatial memories for 3D

reconstruction and embodied perception (Wang & Agapito, 2025; Wu et al., 2025). Recent systems make the write adaptive. VideoTitans directly adapts gradient-driven Titans memory to video forecasting, combining local attention, episodic memory, and persistent tokens (Park et al., 2025). TTT3R recasts the recurrent state of a streaming 3D reconstruction model as fast weights and uses a training-free, confidence-based learning rate to balance new evidence against retention (Chen et al., 2026). Mem3R combines an implicit MLP memory for pose with a fixed-size explicit token state for geometry (Liu et al., 2026). DexWorldModel (CLWM) uses a dual-state TTT memory inside a causal latent world model for long-horizon robotic manipulation (Deng et al., 2026). A different family retains tokens rather than adapting parameters: HERMES manages the multimodal-model KV cache as a training-free hierarchical memory for streaming video understanding (Zhang et al., 2026a). These systems are evaluated primarily by forecasting, reconstruction, control, or question-answering performance. Such evaluation is appropriate for their tasks, but it conflates memory contents with the reader and the task head. We instead audit the adaptive state itself. We do not claim a performance comparison with these systems.

3 A Test-Time Video Memory as a Case Study

3.1 Two nested learning loops

Following Titans (Behrouz et al., 2025b), the model learns at two different timescales (Figure 1). The *inner loop* writes a particular stream into the memory. During one forward pass, each feature supplies a key–value pair, and one associative gradient update changes the memory state, $M_0 \to M_1 \to \cdots \to M_t$. These states are sequence-specific: they carry what has been written so far, are read at the current step, and are reset when a new independent stream begins.

The *outer loop* learns the machinery that makes those online writes useful. After the unrolled inner loop produces a prediction, an outer loss is backpropagated through the write trajectory of the current training window. AdamW then updates the initial memory M_0, the key and value projections, the gate heads, the readout, and the optional Memory-as-Context (MAC) block. It does not directly store the current video's content in the transient states $M_1, \ldots, M_t$. A low outer loss therefore shows that the complete nested system predicts its target. On its own, it does not show that the inner loop wrote information the readout needed. This distinction motivates the state-level audit developed below.

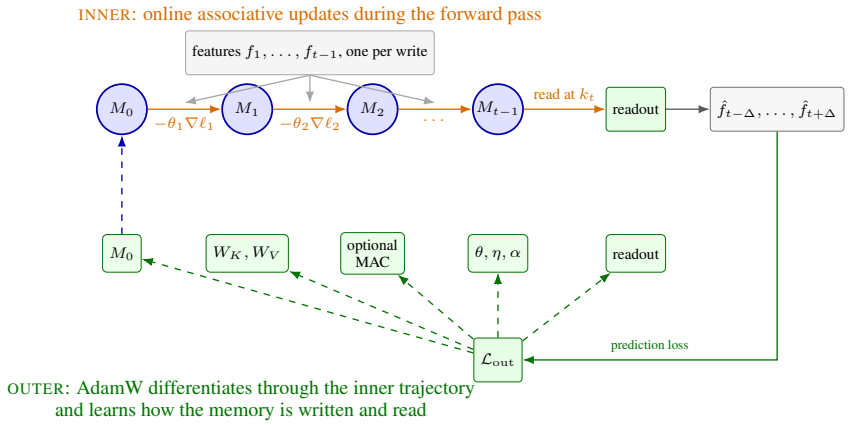

Figure 1: The two nested learning loops. Inner (orange): within a forward pass, the associative loss updates the sequence-specific memory one token at a time. The pre-write state M_{t-1} is read at the current key k_t to predict a feature window around t. Outer (green): the prediction loss is differentiated through the inner updates and trains the shared initialization, projections, gates, readout, and optional context block. The outer loop learns *how* the memory is written and read. The inner-loop state holds *what* the current stream wrote.

Stream and associative memory. A frozen DINOv2 ViT-L/14 encoder (Oquab et al., 2024) maps each frame to a CLS feature in $\mathbb{R}^{1024}$. On video, the stream token f_t is the mean of these features

over a block of consecutive frames (Section 6). Learned projections produce a key k_t, which also serves as the query, and a value v_t,

$$k_t = \mathrm{norm}\big(\mathrm{conv}(\mathrm{SiLU}(W_K f_t))\big), \qquad v_t = \mathrm{norm}\big(\mathrm{conv}(\mathrm{SiLU}(W_V f_t))\big). \tag{1}$$

Here norm is ℓ_2 normalization. The default causal convolution has width four, and both k_t and v_t depend on $f_{t-3}, \ldots, f_t$ (the i.i.d. experiment of Section 6.2 uses no convolution). The writable state M_t is a two-layer MLP whose parameters are the memory and whose output is ℓ_2-normalized. The associative write loss is $\ell_t(M) = \|M(k_t) - v_t\|_2^2 = 2 - 2\cos(M(k_t), v_t) \in [0, 4]$. With surprise S_t, write rate θ_t, momentum η_t, and forget rate α_t (gates in Appendix C), the online update is

$$S_t = \eta_t S_{t-1} - \theta_t \nabla_M \ell_t(M_{t-1}), \qquad M_t = (1 - \alpha_t) M_{t-1} + S_t. \tag{2}$$

All projections, gates, M_0, and the outer readout are learned across videos. Only (M_t, S_t) change within a test stream. We call this within-stream adaptation *test-time optimization*, since it descends the inner loss online. It amounts to *test-time memorization* only if the adapted state retains content of the stream that a later read recovers and that the query alone does not determine.

Outer objectives. The naive recall objective reads the memory at a key after streaming its context and reconstructs its value. It mirrors the associative inner loss at the outer level, and it is the most direct test of whether a written item can be read back. Unlike associative-recall benchmarks, however, it derives the key and the value from the same feature. Section 4 shows why this makes it degenerate. For video we therefore use a windowed objective, which reads before the current write and predicts a symmetric feature window:

$$\mathcal{L}_{\mathrm{win}} = \mathbb{E}_t \sum_{j \in \{-\Delta, \ldots, \Delta\} \setminus \{0\}} w_j \left\| h_j(M_{t-1}(k_t)) - f_{t+j} \right\|_1, \qquad w_j \propto |j|^p, \tag{3}$$

where h_j is a LayerNorm followed by one linear map per offset. For $j < 0$ the target has already been streamed and written. We call this half *causal history reconstruction* rather than recall, because the read is not the only path to the target. The key sees $f_{t-3}, \ldots, f_t$ through the causal convolution, and a copy of f_t already predicts nearby tokens. For $j > 0$ the target has not yet been presented, and this half asks for prediction. A memory that helps there must carry structure that predicts the stream beyond the current token. Excluding $j = 0$ removes exact same-token reconstruction. It does not remove query-only temporal prediction. We score each offset by its copy-relative gain, $\mathrm{gain}_j = \mathbb{E}_t\big[\|f_{t+j} - f_t\|_1 - \|h_j(M_{t-1}(k_t)) - f_{t+j}\|_1\big]$, with both errors averaged over the 1024 feature coordinates. A positive gain means that the model beats a copy of the current token.

4 Can the Model Succeed Without Using Memory?

Low prediction error does not by itself show that the adaptive state carried information from the stream, because the query and the fixed read path may already determine the target. We proceed in three steps. Lemma 1 defines the strongest possible query-only baseline. Proposition 1 gives sufficient conditions under which that baseline can attain zero loss with no context-dependent writes. The causal-history gap asks whether information beyond the query could improve prediction at all. Only after history is useful and reaches the prediction uniquely through the adaptive state can an intervention on the writes, such as disabling or swapping them, attribute success to memorization. These statements are elementary. Their role is to fix what a storage claim must rule out.

The most direct test of storage seems to be recall. At the end of a stream the memory is read at a past key K_i and asked to return a target V_i. Let C be the streamed context, K the query, V its target, and $r_{\lambda,\omega}(C, K)$ the prediction induced by write parameters λ and read parameters ω. The relevant baseline is the best predictor available when the adaptive state is context-independent.

Lemma 1 (Query-only Bayes risk)**.** *If $V \in \mathbb{R}^d$ and $\mathbb{E}\|V\|^2 < \infty$, then over all measurable g,*

$$R_K^* := \inf_g \mathbb{E}\|V - g(K)\|^2 = \mathbb{E}\ \mathrm{tr}\,\mathrm{Cov}(V \mid K), \qquad \textit{attained by } g^*(K) = \mathbb{E}[V \mid K]. \tag{4}$$

The implemented fixed-state read class may have larger risk. Equality requires it to realize the conditional mean.

Proposition 1 (Sufficient condition for write-free degeneracy)**.** *Assume that (i) $V = q(K)$ almost surely for a measurable q, (ii) the architecture admits a parameter setting (λ_0, ω_0) whose adaptive-state trajectory does not depend on C, and (iii) the fixed-state read path of that setting can realize q. Then every non-negative recall loss that vanishes on an exact prediction has global minimum zero, and at least one global minimizer is write-free.*

This is an existence and non-identifiability result. It does not say that every optimum ignores memory or that gradient descent must find one. Sigmoid and softplus gates do not attain exact zero at finite logits, and the architecture-specific statement is in general an infimum approached by increasingly write-free parameters. Nor is the implemented key in equation 1 invertible by construction: SiLU, normalization, convolution, and the unconstrained projection can all discard information. In the trained i.i.d. model of Section 6.2, however, the convolution is absent and W_K is invertible (condition number 406). Only the SiLU and the normalization can then discard information about the feature.

Positive query-only risk does not force memory use. What matters is whether history lowers that risk. Let H denote causal history not already contained in K. Under squared loss, nested conditional expectation gives

$$R^*_K - R^*_{K,H} = \mathbb{E}\,\|\mathbb{E}[V \mid K, H] - \mathbb{E}[V \mid K]\|^2 \geq 0. \tag{5}$$

A positive gap says that history is useful, not that a particular network stored it. The adaptive state must be the only path carrying H, and the trained system must outperform the same read path with state updates disabled. For the implemented ℓ_1 outer loss, Bayes predictors are coordinate-wise conditional medians and the analogous risk gap must be measured empirically. Proofs are provided in Appendix A.

5 Progressive Evidence for Adaptive Storage

We evaluate the memory by answering three questions of increasing strength.

1. **Does the query already solve the task?** Compare against current-feature copying and against query-only predictors that see the receptive field of the key. Measure feature autocorrelation and expose any temporal context embedded in the key.
2. **Does the state depend on the stream?** Track write and forget gates, compare final states produced by different streams, and evaluate the same trained checkpoint with its writes disabled or replaced by those of another stream. Low inner loss alone is not evidence at this level.
3. **Is the dependence content-selective and useful?** Compare written items with unwritten items of the same video, measure retention against lag, and aggregate uncertainty over videos rather than treating correlated tokens as independent samples.

Sections 6.1 to 6.3 address the first two questions, and Section 6.4 applies all three to the video models.

6 Experiments

Setup. We train and evaluate on Nymeria egocentric video (Ma et al., 2024). Of the 1098 extracted sequences, 1065 have a verified 30 Hz frame rate and are used. Unless stated otherwise, a training window contains 64 tokens, each the mean of 50 consecutive frames (1.67 s per token, stride 50), and we use $\Delta = 4$ and $p = 1$. The memory is a two-layer MLP with hidden factor two (about 4.2M parameters). The model is trained in fp32 with AdamW and batch size one, with state carried between consecutive windows and gradients truncated at window boundaries. Each configuration is trained once, with a single seed.

Five-video diagnostic fixture. The principal video evaluation uses one truth video, the video written into memory, and four distractors. Section 6.4 rotates the truth over all five. With the outer parameters frozen, we reset the memory to the learned pre-write state M_0 and replay the full truth

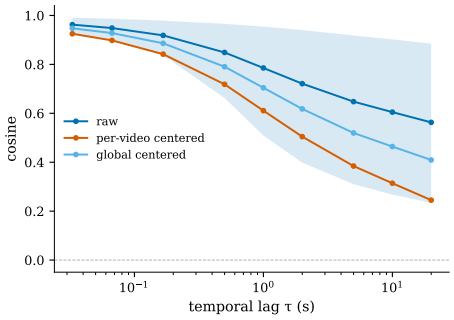

Figure 2: Cosine similarity between DINOv2 features of the same Nymeria video, τ seconds apart, for raw features and after subtracting the global or the per-video mean. The band spans the 10th to 90th percentiles of the raw cosine.

stream through the inner loop to obtain M^*. The four distractor streams are never written into this state. For a key–value pair (k_i, v_i), the paired writing effect is

$$\delta_i = d(M_0(k_i), v_i) - d(M^*(k_i), v_i)\,, \qquad d(a, b) = \|a - b\|_2. \tag{6}$$

The seen/unseen AUC measures how well δ_i separates written tokens of the truth video from unwritten tokens of the four other videos. This measurement queries the adaptive state directly, but its negatives are not matched for scene or video identity.

A second family evaluates the trained temporal head on 32 windows of 64 tokens drawn from the four distractor streams. Each window starts from M_0, whereas training carries the state across windows. We report the far-past gain, $\mathrm{gain}_{-\Delta}$, and the near-offset asymmetry, $\mathrm{gain}_{-1} - \mathrm{gain}_{+1}$, using the copy-relative gain of Section 3. These metrics are evaluated after training, but remain close to the outer objective and can use information in the query path. The random-initialization control is informative only for direct reads of the state, since its temporal head is untrained.

6.1 Correlated video makes copying competitive

Before training a memory, we measure $\cos(f_t, f_{t+\tau})$ at lag τ on the frozen features (Figure 2). At 30 Hz, mean raw cosine falls only from 0.962 at one frame to 0.563 at 20 s. Removing each video's mean lowers the latter to 0.245, showing that much of the long-range similarity is a static scene component. Even at training stride 50, adjacent pooled tokens have cosine 0.869 raw and 0.723 after per-video centering. A copy of f_t thus remains a strong predictor over the whole window.

6.2 Recall loss falls while writing disappears

We remove temporal predictability by training with the recall objective on sequences of 64 DINOv2 features of independently sampled ImageNet validation images (Deng et al., 2009). This approximates the setting of Proposition 1, since the key projection is invertible (Section 4). The objective appears successful: the ℓ_1 recall loss falls from 1.24 to 0.12. Every state diagnostic contradicts this reading. The mean write rate shrinks 200-fold, from 6.4×10^{-2} to 3.0×10^{-4}. The final state equals $0.118\, M_0$ up to a residual of 0.05% of its norm, and $0.118 = 0.9672^{64}$ is the decay produced by the forget gate alone. Written and unwritten items have the same error (0.124 versus 0.123), and on a needle-in-a-haystack check of a second recall-trained run, the same architecture at random initialization retrieves better (Appendix D). The outer objective has reached a nearly write-free solution while its loss was still falling.

6.3 On video, common diagnostics disagree

The past–future objective excludes the exact recall target at $j = 0$. We compare two variants within the same implementation: direct memory, with the momentum-forget update, and a variant inspired by the MAC architecture of Titans. Appendix C gives their update rules. These are diagnostic variants, not complete reproductions of the corresponding systems. In particular, our MAC variant

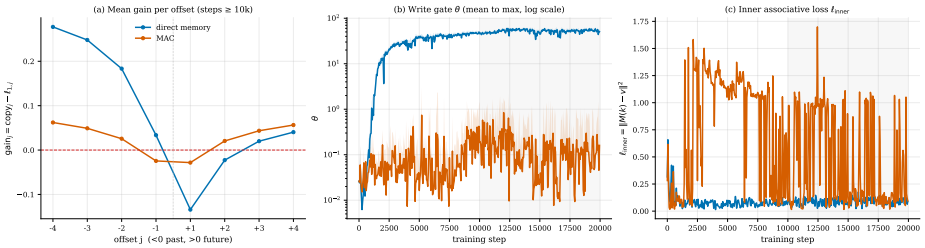

Figure 3: Training-side diagnostics for direct memory and the MAC variant. **(a)** Copy-relative gain by offset, averaged over steps after 10,000. **(b)** Write gate θ on a logarithmic scale, mean per step with a band up to the maximum. **(c)** Associative inner loss. The shaded steps are those averaged in (a) and in Table B.1. No single quantity identifies stored content.

reads its memory at four learned probes that do not depend on the input, whereas Titans queries the memory with a projection of the current segment.

Averaged over training windows after step 10,000, the copy-relative gain of direct memory rises from $+0.034$ at $j = -1$ to $+0.277$ at $j = -4$, whereas its gains at $j = +1, \ldots, +4$ are -0.134, -0.023, $+0.020$, and $+0.040$ (Figure 3a and Table B.1). The resulting past–future gap is consistent with access to causal history, but the causal key convolution can also supply it (Section 6.4). In direct memory, the key contains $f_{t-3}, \ldots, f_t$ even when nothing is written. The MAC key is computed from y_t alone (Appendix C), and earlier features reach its prediction only through the memory.

More importantly, the training-side signals do not identify stored content (Figure 3b,c). Averaged over the same steps, direct memory has a large write rate ($\theta \approx 52$), a negligible learned momentum ($\eta \approx 3 \times 10^{-6}$), and a low, stable inner loss (0.08). The MAC variant has a write rate more than two orders of magnitude smaller ($\theta \approx 0.14$), a strong momentum ($\eta \approx 0.76$), and a fast forget rate. Its mean forget rate, $\alpha \approx 0.14$, halves earlier writes in about five tokens (8 s), against about 28 tokens for direct memory. The MAC variant shows almost no past–future gap. Its inner loss alternates between near zero and about one, with median 0.16 after step 10,000. A low inner loss does not, by itself, indicate a useful write path.

6.4 Attribution controls locate what the state carries

We freeze the outer weights and apply the controls of Section 5 to direct memory and to the MAC variant (Table 1). Each model writes through its own path, and every evaluation window starts from M_0. A stateless baseline completes the set of controls. For each offset, it predicts f_{t+j} as a weighted sum of the four features inside the key window plus a scalar intercept. The same five coefficients serve every feature dimension. They are fit under ℓ_1 on four fixture videos and scored on the fifth, in turn for each video. Copying f_t, copying f_{t-3}, and averaging the window are special cases.

With the default horizon, the readout of direct memory depends on its own state but gains nothing from it beyond the query. With the state written by another video at the same position, the far-past gain falls from $+0.24$ to -0.49 and the near-offset asymmetry turns negative. Disabling the writes is more damaging still, but this arm moves the readout off its training distribution, since M_t leaves M_0 after the first write. Yet a copy of f_{t-3}, the oldest feature inside the key window, matches the far-past prediction to within 0.003. The stateless filter copies the targets at $j = -1, -2, -3$ exactly, since they lie inside the key window, whereas the memory read recovers them only partially. It also matches the model at $j = -4$ and beats it at every future offset. Within a video, the writing effect δ on the last written token exceeds that on the next unwritten tokens by 0.29, and the excess vanishes within a few steps. Direct memory therefore retains item-level content for about one step. This is far shorter than the half-life of 28 tokens set by its forget gate, which suggests that later writes overwrite earlier ones. The prediction of the MAC variant barely depends on its memory. Swapping or disabling its writes changes its gain by at most 0.004 at any offset.

The horizon of the objective accounts for this result. With $\Delta = 4$, three of the four past targets lie inside the key window and the fourth lies one step beyond it. The objective thus asks for little that the query lacks. We apply the same controls to models trained with $\Delta = 10$, whose far-past target

Table 1: Attribution controls on the five fixture videos. The far-past offset is $j = -\Delta$. Gains are computed on 32 windows, with 95% bootstrap intervals over windows where shown. These intervals ignore the correlation of windows within a video. AUCs are means and standard deviations over the five videos. The excess writing effect compares written tokens at a given lag with the next unwritten tokens of the same video, with standard errors of about 0.03. The $\Delta = 10$ model uses offset power $p = 1$. The key of the MAC variant sees only the current step.

Control	Direct, $\Delta = 4$	Direct, $\Delta = 10$	MAC, $\Delta = 4$
Far-past gain over a copy of f_t	$+0.238$	$+0.229$	$+0.045$
over a copy of f_{t-3}	$+0.003$ [-0.05, 0.05]	$+0.184$ [0.12, 0.24]	-0.190
over the stateless filter	-0.011 [-0.06, 0.03]	$+0.111$ [0.06, 0.16]	-0.204
Far-past gain, another video's state	-0.492 [-0.58, -0.42]	-0.336 [-0.43, -0.25]	$+0.042$
Far-past gain, no writes	-1.184	-1.734	$+0.042$
Asymmetry at $\lvert j\rvert = 1$, own/swapped	$+0.153/-0.060$	$+0.068/-0.056$	$+0.003/+0.001$
Excess writing effect, lag 1	$+0.29$	$+0.04$	0.00
lags 2–4	$+0.05$	$+0.06$	$+0.07$
lags 5–10	-0.02	$+0.02$	-0.03
Seen/unseen AUC, five truth videos	0.60 ± 0.25	0.59 ± 0.15	0.61 ± 0.26
at random initialization	0.73 ± 0.17	0.73 ± 0.17	0.70 ± 0.19
Written/unwritten AUC, same video	0.54 ± 0.21	0.51 ± 0.17	0.44 ± 0.13

lies 16.7 s back (Table 1, $\Delta = 10$ column). The model then predicts what the stateless filter does not. At $j = -10$ it beats the filter by 0.11 and a copy of f_{t-3} by 0.18. The advantage over the filter is significant from $j = -7$ to $j = -10$, vanishes at $j = -5$, and reverses near the key window. Swapping the state costs 0.56. As at $\Delta = 4$, this cost shows that the readout relies on its own state, not that the state carries history. Models trained with offset powers 0 and 2 give similar results. At $j = -10$ they beat the copy of f_{t-3} by 0.16 and 0.19, and swapping their state costs 0.54 and 0.57. In all three $\Delta = 10$ models, the associative trace of individual items stays weak, with an excess of at most 0.06 at any lag. The state helps the readout without storing items that can be read back at their own keys.

The seen/unseen AUC registers none of this (Table 1, last three rows). For the truth video used so far, written tokens are read closer to their values than tokens of other videos (Appendix F), which looks like storage. Writing, however, improves the reads of the other videos almost as much (δ of 1.25 on the other videos against 1.50 on the written one). With each fixture video in turn as the truth stream, the AUC ranges from 0.23 to 0.92 for direct memory, and this video gives the highest value. The random-initialization control is higher on average for every model. Written and unwritten tokens of the same video are not separated either. The seen/unseen AUC therefore tracks scene identity rather than item storage.

6.5 Training spacing exposes a conflict between metrics

A further sweep varies the training regime, namely the window length in tokens and the wall-clock spacing between consecutive tokens (Figure 4). Every trained model is then probed on the same five-video fixture, at stride 50 and with 64-token windows. In effect, the sweep measures how each training regime transfers to one evaluation spacing. The random-initialization arm is identical in every cell (seen/unseen AUC of 0.971), which confirms that only training varies. All cells use the truth video that gives the highest AUC in Section 6.4.

At window length 32, the far-past gain (Figure 4, middle) rises from -0.13 for the model trained at stride 1 to $+0.21$ for the model trained at stride 100. The best cell is length 128 at stride 50, with gain $+0.29$. Only this cell exceeds the gain of a copy of f_{t-3} on the same windows (0.235), by 0.05 in a single run without an interval. The near-offset asymmetry also rises with training spacing (right). Several causes are confounded in this trend. At 30 Hz adjacent frames are nearly identical (Figure 2). A copy is then a strong predictor during training, and the model may learn little beyond it. A model trained at stride 1 is also probed out of distribution. Its head for $j = -4$ was trained to predict the feature 0.13 s in the past from single frames, and it is asked for the feature 6.7 s in

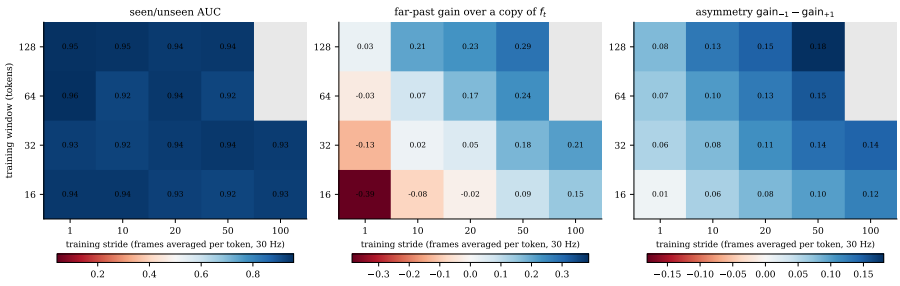

Figure 4: Five-video metrics for models trained over a grid of window length and sampling stride. Every cell is probed at stride 50 with 64-token windows. **Left:** seen/unseen AUC. **Middle:** far-past gain over a copy of f_t, where a copy of f_{t-3} reaches 0.235 on the same windows. **Right:** near-offset past–future asymmetry. AUC is highest for a model trained at stride 1, whereas the temporal metrics improve with the training stride. Gray cells have no run.

the past from 50-frame averages. Finally, the wall-clock span of a training window grows with the stride. The sweep does not separate these causes.

The seen/unseen AUC (left) varies little (0.919 to 0.958), stays below the random-initialization control in every cell, and ranks the training regimes differently from the temporal metrics.

7 Discussion and Limitations

What is not established. The video experiments use one encoder, one dataset, five probe videos, and one training seed per configuration. Our stateless baseline is a linear filter fit on four videos. A nonlinear predictor trained on the full corpus could recover part of the far-past benefit. The state of the $\Delta = 10$ model helps the readout, but it does not store items that can be read back at their own keys. Disabling the writes of a trained checkpoint moves its readout off distribution, and we rely on the state swap for that comparison instead. Gain intervals resample windows rather than videos and understate the uncertainty. We therefore claim neither a working long-term episodic memory nor superiority over the systems of Section 2.

Practical implication. A credible storage claim should report four comparisons: against a query-only predictor that sees the receptive field of the key, against the same checkpoint with its state swapped across streams, between content-matched written and unwritten items, and across several written streams. The objective must also reach beyond the receptive field of the key, since otherwise a stateless predictor can meet it. Retention should be indexed by time and interference, with uncertainty aggregated over independent videos. This standard applies, in principle, whether the state is a fast-weight MLP, a recurrent latent, or a compressed KV cache. Only the appropriate intervention changes.

8 Conclusion

Test-time optimization can succeed without test-time memorization, and on correlated video the gap between the two is large enough to reverse the conclusion of a standard diagnostic. This work makes that gap both formal and measurable. The query-only risk and the write-free proposition identify when an objective cannot establish memory use. The recall-collapse experiment shows that gradient-based training converges toward this failure. The video study shows that, with the default horizon, every positive diagnostic is explained by the key window or by scene identity. When the objective reaches beyond the key window, the model outperforms a stateless filter while relying on its own state, and the standard diagnostics do not reveal this advantage. The contribution is not a new memory but a protocol against false attribution. It lets one compare adaptive architectures by whether their state carries information from the current stream.

Reproducibility Statement

Section 3 and Appendices C and E specify the architecture, update rules, gates, and training schedule. Appendix A gives complete proofs under the stated assumptions. Code for training, the probes, and the controls is provided as supplementary material.

Ethics Statement

We use the Nymeria dataset under its license terms. It contains egocentric video of human participants. We process only frozen DINOv2 features of its preview videos, and we release no frames, identities, or derived personal data.

AI Use Statement

Generative AI tools assisted with code implementation and review, analysis scripts, literature organization, mathematical checking, manuscript restructuring, and prose editing. The authors designed the experiments, validated the code with runnable checks, selected the scientific claims, verified the mathematical statements and citations against primary sources, and take responsibility for the contents of the submission, including text, claims, and artifacts produced with the aid of generative AI.

References

Daniel J. Amit, Hanoch Gutfreund, and Haim Sompolinsky. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Physical Review Letters*, 55(14):1530–1533, 1985.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Ali Behrouz, Zeman Li, Praneeth Kacham, Majid Daliri, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. ATLAS: Learning to optimally memorize the context at test time. *arXiv preprint arXiv:2505.23735*, 2025a.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. It's all connected: A journey through test-time memorization, attentional bias, retention, and online optimization. In *International Conference on Learning Representations (ICLR)*, 2026.

Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the "video" in video-language understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Aydar Bulatov, Yuri Kuratov, and Mikhail S. Burtsev. Recurrent memory transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3R: 3D reconstruction as test-time training. In *International Conference on Learning Representations (ICLR)*, 2026.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Yueci Deng, Guiliang Liu, and Kui Jia. DexWorldModel: Causal latent world modeling towards automated learning of embodied tasks. *arXiv preprint arXiv:2604.16484*, 2026.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673, 2020.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Alex Graves, Greg Wayne, Malcolm Reynolds, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538:471–476, 2016.

John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Conference on Language Modeling (COLM)*, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning (ICML)*, 2020.

Teuvo Kohonen. Correlation matrix memories. *IEEE Transactions on Computers*, C-21(4):353–359, 1972.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. BABILong: Testing the limits of LLMs with long context reasoning-in-a-haystack. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Jie Lei, Tamara L. Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Zeman Li, Ali Behrouz, Yuan Deng, Peilin Zhong, Praneeth Kacham, Mahdi Karami, Meisam Razaviyayn, and Vahab Mirrokni. TNT: Improving chunkwise training for test-time memorization. In *International Conference on Learning Representations (ICLR)*, 2026.

Changkun Liu, Jiezhi Yang, Zeman Li, Yuan Deng, Jiancong Guo, and Luca Ballan. Mem3R: Streaming 3D reconstruction with hybrid memory via test-time training. *arXiv preprint arXiv:2604.07279*, 2026.

Lingni Ma, Yuting Ye, Fangzhou Hong, Vladimir Guzov, Yifeng Jiang, Rowan Postyeni, Luis Pesqueira, Alexander Gamino, Vijay Baiyya, Hyo Jin Kim, Kevin Bailey, David Soriano Fosas, C. Karen Liu, Ziwei Liu, Jakob Engel, Renzo De Nardi, and Richard Newcombe. Nymeria: A massive collection of multimodal egocentric daily motion in the wild. In *European Conference on Computer Vision (ECCV)*, 2024.

Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. Leave no context behind: Efficient infinite context transformers with Infini-attention. *arXiv preprint arXiv:2404.07143*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Young-Jae Park, Minseok Seo, and Hae-Gon Jeon. VideoTitans: Scalable video prediction with integrated short- and long-term memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning (ICML)*, 2021.

Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

Enxin Song, Wenhao Chai, Guanhong Wang, et al. MovieChat: From dense token to sparse memory for long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Simeng Sun, Kalpesh Krishna, Andrew Mattarella-Micke, and Mohit Iyyer. Do long-range language models actually use long-range context? In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): RNNs with expressive hidden states. In *International Conference on Machine Learning (ICML)*, 2025.

Hengyi Wang and Lourdes Agapito. 3D reconstruction with spatial memory. In *International Conference on 3D Vision (3DV)*, 2025.

Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. MeMViT: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yuqi Wu, Wenzhao Zheng, Sicheng Zuo, Yuanhui Huang, Jie Zhou, and Jiwen Lu. EmbodiedOcc: Embodied 3D occupancy prediction for vision-based online scene understanding. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations (ICLR)*, 2025.

Haowei Zhang, Shudong Yang, Jinlan Fu, See-Kiong Ng, and Xipeng Qiu. HERMES: KV cache as hierarchical memory for efficient streaming video understanding. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026a.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T. Freeman, and Hao Tan. Test-time training done right. In *International Conference on Learning Representations (ICLR)*, 2026b.

A PROOFS

Proof of Lemma 1. For any measurable $g(K)$, add and subtract the conditional mean. Orthogonality of conditional expectation makes the cross term vanish, so

$$\mathbb{E}\|V - g(K)\|^2 = \mathbb{E}\|V - \mathbb{E}[V \mid K]\|^2 + \mathbb{E}\|\mathbb{E}[V \mid K] - g(K)\|^2. \tag{7}$$

The first term is $\mathbb{E}\,\mathrm{tr}\,\mathrm{Cov}(V \mid K)$. The second is non-negative, and it vanishes exactly when $g(K) = \mathbb{E}[V \mid K]$ almost surely.

Proof of Proposition 1. Under the write-free parameters (λ_0, ω_0) of assumption (ii), the adaptive-state trajectory is independent of C. Read realizability gives $r_{\lambda_0,\omega_0}(C, K) = q(K) = V$ almost surely. Almost every sample then has zero loss. Non-negativity makes zero globally minimal, and the construction provides a write-free minimizer. It proves existence, not uniqueness or optimizer convergence.

For a finite horizon with sigmoid/softplus gates, suppose a parameter sequence makes $\sup_t \theta_t \to 0$ and $\sup_t \alpha_t \to 0$ while gradients and intermediate states remain bounded. Induction through equation 2 gives $S_t \to 0$ and $M_t \to M_0$. Continuity and dominated convergence make the risk approach that of the fixed-state read. If q lies in the appropriate closure of that read class, the full objective has infimum zero even when no finite gate logit realizes an exact hard-off state.

Causal-history gap. Applying Lemma 1 to K and (K, H) gives

$$R^*_K = \mathbb{E}\|V - \mathbb{E}[V \mid K]\|^2, \tag{8}$$

$$R^*_{K,H} = \mathbb{E}\|V - \mathbb{E}[V \mid K, H]\|^2. \tag{9}$$

Because $\mathcal{F}_K \subseteq \mathcal{F}_{K,H}$, where $\mathcal{F}_X$ denotes the σ-algebra generated by X, the nested orthogonal-projection identity yields

$$R^*_K = R^*_{K,H} + \mathbb{E}\|\mathbb{E}[V \mid K, H] - \mathbb{E}[V \mid K]\|^2, \tag{10}$$

which proves equation 5. The query-only Bayes risk is not a global lower bound when history is available. For example, let K_i be constant and $V_i = Z$ for a sequence-level binary latent Z. Query-only risk is positive, but a causal state that retains V_{i-1} predicts all later V_i exactly.

B FULL PER-OFFSET WINDOW RESULTS

Table B.1: Training-side copy-relative gain_j (mean over steps after 10,000) for the two variants. The table is reported for completeness. The main text uses it only as a diagnostic of disagreement.

	Past offsets j				Future offsets j			
Variant	-4	-3	-2	-1	$+1$	$+2$	$+3$	$+4$
Direct memory	0.277	0.248	0.183	0.034	-0.134	-0.023	0.020	0.040
MAC variant	0.062	0.049	0.026	-0.024	-0.028	0.020	0.043	0.056

C MEMORY VARIANTS

All variants share the key and value maps of equation 1 and the associative loss $\ell_t(M) = \|M(k_t) - v_t\|_2^2$. The output of M is ℓ_2-normalized, so that $\ell_t = 2 - 2\cos(M(k_t), v_t) \in [0, 4]$. The gates are linear heads on the current input,

$$\theta_t = \mathrm{softplus}(w_\theta^\top f_t + b_\theta), \qquad \eta_t = \sigma\big((w_\eta^\top f_t + b_\eta)/\tau\big), \qquad \alpha_t = \sigma\big((w_\alpha^\top f_t + b_\alpha)/\tau\big), \tag{11}$$

with temperature $\tau = 4$. The variants differ in how the gradient becomes a write.

SGD-forget. The plainest rule takes one gradient step and lets the forget gate shrink the old weights,

$$M_t = (1 - \alpha_t) M_{t-1} - \theta_t \nabla_M \ell_t(M_{t-1}). \tag{12}$$

Trained with it, direct memory reaches nearly the same far-past gain and asymmetry as with momentum (0.236 and 0.158, against 0.238 and 0.153).

Momentum-forget (direct memory). The default rule is the Titans update of equation 2, which accumulates the surprise in S_t before committing it. It reduces to equation 12 at $\eta_t = 0$. The readout uses the pre-write state, $\hat{f}_{t+j} = h_j(M_{t-1}(k_t))$, as in equation 3. In the trained model the learned momentum is negligible ($\eta_t \approx 3 \times 10^{-6}$), so the rule acts as equation 12.

MAC variant. Each pooled step reads the memory at four learned probes u, $r_t = M_{t-1}(u/\|u\|)$. Self-attention then mixes 16 persistent tokens P, the retrieved tokens, and the current feature, and keeps the output at the position of f_t, $y_t = [\mathrm{Attn}(P \,\|\, r_t \,\|\, f_t)]_{f_t}$. The inner loop writes y_t in place of f_t with the momentum-forget rule. A sigmoid gate built from the post-write read gives the step output $o_t = y_t \odot \sigma(W_g \,\mathrm{LayerNorm}(M_t(k(y_t))) + b_g)$, which the readout consumes. The key is computed from the single vector y_t, so the causal convolution of equation 1 sees no earlier step.

D Controlled Needle-in-a-Haystack Recall Check

We separately constructed a controlled associative-recall check on static ImageNet features. An episode inserts a target image (the needle) into a stream of distractor images, replays the stream to obtain M_T, and queries the target key at the end. Each episode is paired with an otherwise identical stream in which the needle is absent. The protocol scores the paired change $\cos(M_T(k), v) - \cos(M_0(k), v)$ and the present-versus-absent AUC, with raw-cosine, sliding-window kNN, pre-write M_0, and random-initialization controls. The protocol can vary the number and length of needle bursts, the lag from the last occurrence to the query, exact-instance versus held-out-instance queries, and same-class versus different-class distractors.

For a second recall-trained run with the configuration of Section 6.2, the replay covered 26 cells of the protocol, with sequences of 10 to 250 items. The random-initialization control separated present from absent needles in all 26 cells (AUC 0.744 to 1.000), with a positive paired change. The trained model was significant in only 6 cells, all at 250 items and short lags. Their AUC was 0.67 to 0.74, but writing lowered the cosine at the needle key, to about 0.35 against 0.63 before writing. This is consistent with the main audit: outer training closed the write path and removed retrieval already present at initialization. A correlated pseudo-video assembled from ImageNet likewise did not separate the trained memory from its random-initialization control and is not used to support the video-storage claim.

E Implementation Details

The outer objective is differentiated through the sequence of inner updates. Each inner step is evaluated functionally and differentiated with its second-order terms retained, allowing the outer optimizer to train the key/value projections, gates, initial memory, and readout. State carries between consecutive windows from the same video, while the second-order graph is truncated at window boundaries. This is within-video memorization: both M and the surprise state S are reset at a new independent stream.

All main video runs use DINOv2 ViT-L/14 CLS features of width 1024, mean pooling stride 50, window length 64, a two-layer GELU memory with hidden factor two, linear data-dependent gates, sigmoid temperature four, and momentum-forget updates. The outer optimizer is AdamW under a warmup–cosine schedule, batch size one, fp32, for 20,000 steps. A direct-memory run takes about one hour on one NVIDIA RTX A5000, and a MAC-variant run about seventeen hours, because its loop is sequential per step. The controls of Section 6.4 replay the exact recurrent inner loop with the same memory operations as the training forward pass, so that the written and the read streams can differ. Their unmodified arm reproduces the forward pass of each checkpoint exactly. The stateless filter is fit by iteratively reweighted least squares under ℓ_1, leave-one-video-out, with its

five coefficients shared across the 1024 feature dimensions. Unconstrained linear maps from the key window or from the key itself, fit on four videos, do worse than a copy on the held-out video and are not reported. The $\Delta = 10$ models come from a sweep over the readout horizon, $\Delta \in \{1, 4, 10\}$, and the offset power, $p \in \{0, 1, 2\}$, and follow the default recipe otherwise.

F Seen and Unseen Tokens of the Truth Video

Figure F.1 shows the seen/unseen separation behind the AUC of Section 6.4 for the truth video of the main text.

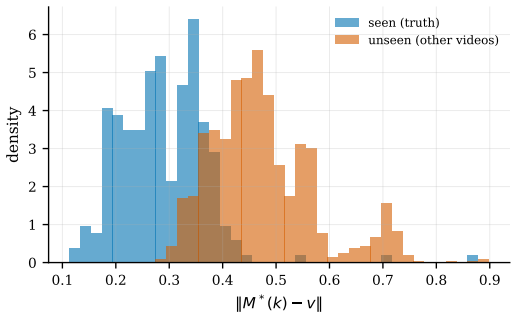

Figure F.1: Distance between the memory read and the value after streaming the truth video, for its written tokens (seen) and for tokens of the four other videos (unseen). The separation looks like storage, but this truth video gives the highest AUC of the five, and written and unwritten tokens of the same video are not separated (Table 1).