Androids Dream of the Sheep They Saw That Day: Test-Time Memory and Dream Consolidation in a Small Language Model

Gianni Sanrochman
gsanrochman@proton.me

November 2026 (draft)

Abstract

We build and measure a complete learn-by-day, consolidate-by-night loop in a 134M-parameter language model. During the day the model reads a text stream with its weights frozen; new facts live only in a fast tier, a cache of the previous window's attention keys and values. At night the model is cued with answer-free prompts about the day and completes them from that cache. These *dreams*, filtered by self-consistency voting, are the only training data for a set of sparse lifetime tables; a gate on held-out loss and on recall of everything committed before can roll the write back. No ground-truth answer enters consolidation. Four results stand out. (1) A retrieval curriculum that keeps every answer out of the attention window, the standard guard against leakage, trained the model out of in-context copying (31.8% to 3% first-token accuracy with the fact in context), silently confounding four rounds of memory-architecture experiments. (2) Every newly added memory module we tried (a TTT/Titans-style fast-weight matrix, a Nested-Learning continuum memory, per-token write strength, an uncompressed slot memory) failed at *addressing* rather than storage; reusing the backbone's own attention through a Transformer-XL cache raised one-window-back recall from 4.3% to 94.1%. (3) Voted dreams consolidate at 81% of the ground-truth control and preserve old knowledge under later learning. (4) The first full loop cleared its pre-registered bars only at the margin, because the cache could hold a fact without the model being able to answer a cue from it (27% at realistic density). Short training on the cue format with held-out subjects and per-sample random answers raised cache dreams to 100%; tuning only the attention query and key projections, against real post-sleep tables, does so within 0.013 of the untuned model's val loss, and the loop then learns 57% of a day's facts (0% without sleep). Two label-free checks on rehearsal, whether the tables change a dream's answer and whether it is stable across nights, raise re-dreamed old facts from 27 to 77% correct. Restricting each night's write to the table rows that the night's dreams use far more than ordinary text then lets the model keep 81% of a day's facts after two more days (40% with neither, 94% when sleep trains on the true answers; means of three seeds); splitting sleep into one small sleep per window, so that consolidation runs during the day, fails. All thresholds were set before each run, and every probe is reported against the model's own in-context ceiling.

1 Introduction

A deployed language model that learns something new must either keep it in context, which is bounded, or change its weights, which interferes with what it already knows. Complementary learning systems theory [11] describes how brains split the problem: the hippocampus records episodes quickly, and sleep replays them into slow cortical memory. Recent architectures import

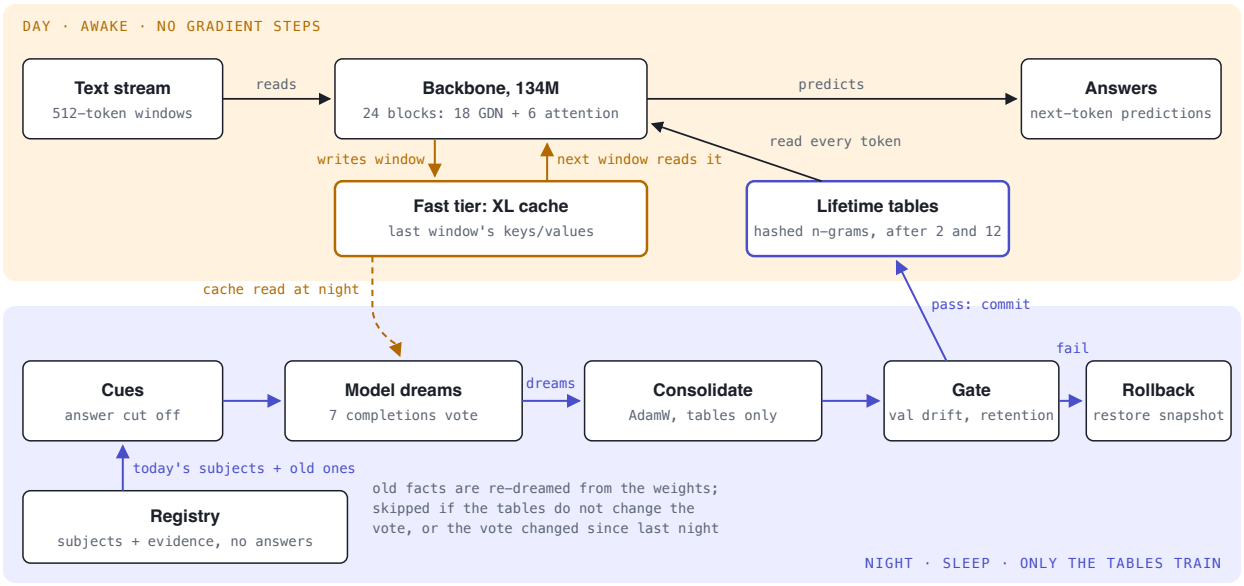

Figure 1: The loop. Day: no weight changes; the backbone reads a stream in 512-token windows, each attention block also sees the previous window through the XL cache, and every token reads the lifetime tables. Night: cues with the answer cut off are completed seven times from the cache, the completions vote, and only the voted dreams train the tables. Old facts are re-dreamed from the weights and filtered without the answer (Section 4.6). A gate on held-out loss and old recall commits the write or restores the snapshot.

the fast half of that story as test-time learning [15, 2, 1]. The slow half, and the loop between them, is less explored in language models.

We build the whole loop at small scale and measure each stage separately before measuring it end to end. Our model, *accretion*, has three tiers:

- a **fast tier** that holds what the model read today without any gradient step;
- **lifetime tables**, sparse hashed embeddings injected into the residual stream and written only during sleep;
- a **sleep job** that writes the tables from the model's own *dreams*, rehearses older knowledge by re-dreaming it, and is guarded by an external gate with rollback.

Figure 1 shows one day and one night.

The paper's contributions are as much about measurement as about the system. Two of our intermediate conclusions were wrong for the same reason: a pass bar was set without first measuring what the unassisted backbone could do with the answer in plain view (Section 3). We report those errors, because the failure mode they expose is general.

2 System

2.1 Backbone

A 24-block decoder, $d = 640$, GPT-2 tokenizer, trained on FineWeb-10B [12]. Eighteen blocks are Gated DeltaNet [21]; every fourth block is grouped-query attention with QK-norm and no

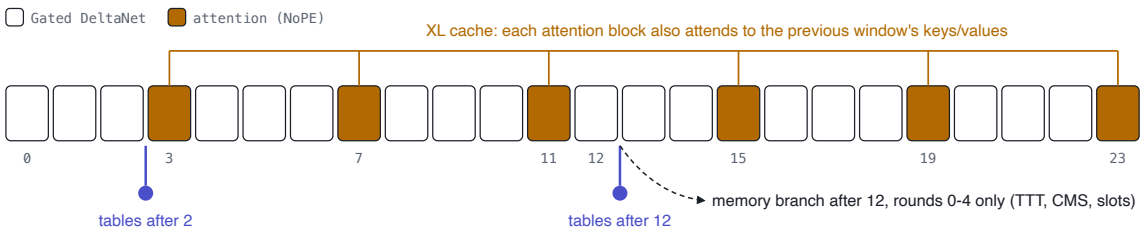

Figure 2: Where the memories sit. Eighteen Gated DeltaNet blocks (white) and six attention blocks (every fourth). The XL cache attaches to the attention blocks; the lifetime tables inject into the residual stream after blocks 2 and 12. The separate memory branch after block 12 was used only in rounds 0 to 4.

positional encoding. An ablation queue at 2B tokens selected this hybrid over dense attention (val 3.259 vs 3.286) and rejected multi-token prediction and a content-blind Engram-style table. Optimizer: Muon for matrices, AdamW elsewhere, warmup-stable-decay schedule. Figure 2 shows where each memory attaches.

2.2 Fast tier

We tested five fast tiers, each trained jointly with the backbone on a retrieval curriculum (Section 3.1): a TTT-Linear fast-weight matrix with learned initial state and momentum [15, 2], gated into block 12 (MAG); a three-level continuum memory system at update periods 32/128/512 tokens [1]; backpropagation through two consecutive windows plus an InfoNCE readback loss; per-token write strength; and an uncompressed slot memory that stores every token's key and value and reads them by multi-head attention. The tier that works is the simplest: a Transformer-XL cache [6] (no new parameters beyond one bias per key-value head). Each attention block also attends to the previous 512-token window's keys and values, placed before the current window; a learned per-head bias on cached logits marks them as past (Section 4.1).

2.3 Lifetime tables and sleep

Lifetime memory is a pair of hashed n-gram tables in the style of Engram [4], injected after blocks 2 and 12. A sleep cycle trains only the tables (AdamW, 12k steps) on restructured documents about the day's facts. Two safeguards come from earlier failures. *Replay*: 30% of each batch rehearses previously committed facts. *Gate*: a write is committed only if held-out val loss stays within +0.05 of the original baseline (not of the previous cycle, which let small drifts compound to +0.09) and previously committed facts keep at least half of their recall; otherwise the pre-sleep snapshot is restored. The model never certifies its own edit.

2.4 Dreaming

In the ground-truth version of sleep the training documents contain the true answers. In the dreaming version they never do. For each fact the registry holds only a cue index (subject and relation). Each of seven restructured phrasings is cut just before the answer, producing a cue such as `Q: Who founded Quonsirn?\nA:`; the object is replaced by a sentinel before templating, so it cannot leak (tested). The model completes each cue greedily. The day's facts are dreamed with the day's memory available; older facts are re-dreamed from the current weights alone. With *consensus*,

Table 1: First-token accuracy, fact in context, memory off, by distance in tokens between statement and question (403 probes).

checkpoint	0	16	32	64	128	256	∼450	no fact
base (no curriculum)	31.8	26.1	22.3	12.9	7.9	2.7	2.7	2.5
old curriculum	3.0						4.2	5.0
with near probes	95.0	95.5	95.5	95.5	95.0	94.8	89.1	6.7

the seven dreams vote on the answer and every cue is completed with the winner, which also strips invented detail. Wrong dreams are consolidated anyway and counted as false memories.

3 Benchmarks and calibration

3.1 Cross-window recall

A quarter of training lanes carry a synthetic stream: novel facts with random two-syllable names ("the mayor of Blafro is Kresut"), filler sentences, and probes asking for a fact one or more windows after its last mention. An anti-leak invariant guarantees that no probe shares a window with any mention of its fact. The metric is first-token accuracy on the answer, with memory carried across windows minus the same checkpoint with memory reset each window, bucketed by the probe's age in windows.

3.2 The ceiling we did not measure

We later measured the same probes with the statement placed back in context and memory off (Table 1). Before curriculum training the backbone copies a novel name from a nearby sentence 31.8% of the time, falling to chance by 256 tokens (2.7%, against 2.5% with no fact in context). After curriculum training it does not copy at all: 3.0%, equal to the no-fact floor. Because the anti-leak rule kept every answer out of attention, and the untrained memory supplied nothing, the best policy the model could learn was to guess a plausible name. Every memory arm trained on that curriculum was feeding an answer path the curriculum had switched off. Re-asking 15% of facts inside their own window ("near" probes, logged separately and excluded from scoring) restored the circuit to 95% and extended it to 89% at 450 tokens.

3.3 Lifetime benchmark

Three disjoint corpora of 60 facts each are consolidated in sequence; after every cycle every earlier corpus is re-probed with a light paraphrase whose in-context ceiling is 100%. Answers come from a list of 10 names, so random guessing scores 10%. Gate-off runs are mandatory: a gate that rolls everything back forgets nothing and learns nothing.

4 Results

4.1 The fast tier fails at addressing

Table 2 lists every fast-tier round. Rounds 0 to 2 are confounded by the curriculum. Round 3 is the first fair test of the fast-weight matrix: +3.5pp. A diagnostic that ranks the stated fact's tokens

Table 2: Fast-tier rounds. Memory gain = first-token accuracy on facts stated one window (512 tokens) earlier, memory carried minus reset, same checkpoint (423 probes; SE of a difference ≈2.3pp). Pre-registered pass: ≥+10pp.

round	fast tier	gain	note
0	TTT-Linear matrix	+1.4	
1	3-level CMS	−0.2	slow levels pruned; val 3.94
1	matrix + 2-window BPTT + readback	−0.9	readback 1.01 vs chance 4.16
2	+ per-token write strength	−1.7	
curriculum fixed: in-context copying restored			
3	per-token matrix	+3.5	read rank 18/512
4	slot memory (no compression)	+1.2	gate 1.00; fact mass 3.9% vs 2.7%
5	Transformer-XL cache	**+89.8**	94.1% vs 4.3%; val +0.16
6	XL + cache bias + denser, wrapped curriculum	+88.4	val 3.496; bias ≈0
7c	round 5 + cue tuning (q/k only)	+88.2	val 3.511

among all keys of the previous window shows why: the query at the question finds the fact only diffusely (median best rank 18 of 512) and the read returns a blend.

Round 4 removed compression entirely. The slot memory's gate opened fully (1.00) and its output projection's norm grew about 300-fold, so the model used the branch heavily, but not for facts: attention mass on the fact's tokens was 3.9% against 2.7% chance. A freshly initialized memory must learn retrieval from sparse probe gradients and instead learned to inject generic context. The backbone's own attention already solves the task at 89 to 95% in context. Round 5 reuses it, and recall one window back rises to 94.1% against 4.3% without the cache. Ages 2 and beyond stay at zero, as expected for a one-window cache.

The cache has a cost. With the cache carried, FineWeb val is 3.498 against 3.342 for the round-3 model; the gap persists under every evaluation protocol. Our hypothesis was that NoPE attention infers order from the causal mask, which an always-visible cache breaks. Round 6 tested it with a learned per-head bias on cached logits, plus a denser curriculum (50% fact lines against 35%) with facts wrapped in document-style framings. Recall one window back held (+88.4), but val did not move (3.496) and the learned bias stayed near zero, so the hypothesis is not supported. The gap remains open; we leave it as a limitation.

4.2 Sleep, replay and interference

On the earlier memory-native checkpoint, consolidation without replay erased corpus A almost entirely after two later cycles (kept 12.5% homogeneous, 0% on varied domains). With 30% replay, corpus A kept 77.8% and 76.5%, and total retained knowledge rose 2× and 6 to 8×. A 2×2 places the interference in shared projection capacity rather than bucket collision: varied corpora with 1.3% bucket overlap still forgot completely, while homogeneous corpora with 56% overlap retained more; freezing the shared projections delayed collapse by one cycle and crippled new learning. Replay works because it keeps old facts in the gradient that shapes the shared projections.

A caution on the gate: the +0.05 drift budget sits at run-to-run noise. The same ground-truth configuration committed its first cycle at +0.045 in one run and rolled it back at +0.051 in another (the tables are randomly initialized). We therefore report gate-off runs as the primary measurement.

Table 3: Sleep on ground truth vs on dreams (memory-native checkpoint, day held in context, gate off, 3×60 facts, 30% replay).

	truth	raw dreams	voted dreams
facts retained of 180	∼98	∼48	∼79
corpus A kept after 2 more cycles	81%	75%	58%
val drift (budget +0.05)	+0.041	+0.058	+0.043
day dream accuracy	–	56–67%	77–100%

Table 4: Full loop on the round-5 model, gate off. Recall is with the fast tier empty.

	truth	dreams, day in context	dreams, day in cache
recall with no sleep	0–3%	0–3%	0–3%
corpus A after its sleep	58.3%	43.3%	10.0%
corpus A after 2 more days	58.3%	18.3%	5.0%
facts retained of 180	∼101	∼68	∼24
day dream accuracy	–	80–98%	15–35%
val drift	+0.010	+0.018	+0.018

4.3 Dreaming

Table 3 compares sleep on the true answers with sleep on the model's own dreams. Raw dreams preserve old knowledge nearly as well as the truth (corpus A keeps 75%) but consolidate only 49% as many facts and drift past the budget. The dreams look like dreams: the core fact is usually right, details are invented and inconsistent ("founded by Nimbleton in 1855", then "in 1844"), and some are false ("the capital of Lutvay is Lutvay"). Consensus voting fixes both failures: 81% of the control's retained facts (∼79 against ∼98 of 180) and drift within budget (+0.043). Re-dreamed old facts are only as accurate as current recall, so mis-remembered facts are rehearsed as faithfully as correct ones.

4.4 The full loop

On the round-5 model, 60 facts are packed whole into six 512-token windows with filler and streamed through the model with the cache carried. Each fact is dreamed from the cache state right after its window, with its text no longer in context. Voted dreams train the tables; recall is probed on the following days with the cache empty (Table 4).

The loop clears its pre-registered bars (learning ≥+10pp over no sleep; corpus A keeps ≥50%; drift ≤+0.05), but at the threshold: six facts learned, three kept (60 facts per corpus, one seed). Every stage behaves as measured in isolation. Sleep turns between half and two thirds of correct dreams into lasting recall in every arm (10/15, 22/35, 20/29 from the cache; 43/80 with the day in context), so the bottleneck is the first stage. Dream accuracy from the cache falls with fact density (voted: 70, 55, 47, 33, 27% at 1, 2, 3, 5 and ∼10 facts per window), and even one fact per window reaches only 70% on framings the model never trained on.

Round 6 retrained the backbone on denser, document-wrapped facts to close that gap. It made it worse: voted cache dreams fell to 52, 37 and 8% at 1, 3 and packed facts per window, and the loop failed three of its four bars (corpus A +8.3pp, kept 40%). Denser retrieval training teaches the model to answer *probes* from the cache; it does not teach it to answer *cues*.

Table 5: Full loop after cue tuning (day streamed through the cache, voted dreams, gate off, 3×60 facts, one seed). "Truth" trains the round-7 model's sleep on the true answers. Bars: val ≤3.52, A learned ≥+20pp over no sleep, A kept ≥50%, drift ≤+0.05.

	r5	r7 all	r7 truth	r7b q/k	r7c q/k + tables
val, cache carried	3.498	3.673	3.673	3.508	**3.511**
cache dreams, packed	27%	100%	–	100%	100%
corpus A after its sleep	10.0%	56.7%	58.3%	56.7%	56.7%
corpus A after 2 more days	5.0%	38.3%	55.0%	18.3%	28.3%
A kept	50%	67.6%	94.3%	32.3%	**50.0%**
facts retained of 180	∼24	∼79	∼100	∼46	∼56
day dreams, nights 1/2/3		100/95/88	–	–/85/66	100/93/80
replay dreams, nights 2/3		60/42	–	28/23	47/24
val drift	+0.018	+0.022	+0.020	+0.021	+0.024

4.5 Cue tuning

We therefore trained the dreaming step directly. Each sample is two windows. The first holds 1 to 12 fact documents in the format the day stream uses, padded with FineWeb filler to 512 tokens; the second holds the night's cues for those facts, each followed by its answer, with the first cue at position 0 so that nothing of the first window is in context. The loss is cross-entropy on the answer tokens only, plus the ordinary language-modeling loss on a FineWeb batch at weight 1, and gradients flow through the carried cache. Two choices keep this from being memorization. The subjects of the three evaluation corpora are held out of training (241 of the 400 possible names remain), and every answer is drawn fresh for every sample, so no subject is tied to any object in the weights; the only thing that can be learned is to read the answer from the cache. Leak controls on the evaluation corpora confirm it: voted dream accuracy is 100% with the right cache, 13 to 20% with another day's cache, and 0 to 2% with no memory (chance is 10%).

Table 5 lists the variants. Tuning all weights (round 7, lr 5×10^{-5}, 1600 steps) gives perfect cache dreams at every density and the strongest loop, but it costs language modeling (val carried 3.673, against a pre-registered ceiling of 3.52). Tuning only the query and key projections of all attention layers (2.9M parameters) keeps val within budget (3.508) and keeps the cache dreams, but corpus A keeps only 32%: dreams of the current day decay from 85 to 66% across nights. The cause is that the cue read was tuned with the lifetime tables empty, and after sleep they are not. Round 7c continues the query/key tuning for 800 steps with the tables sampled from three real post-sleep states (built from training-name corpora only). It passes all four pre-registered bars: val 3.511, cache dreams 100% at every density, corpus A learned at 56.7% against 0% without sleep, kept at exactly 50.0% after two more days, drift +0.024.

The remaining loss is in rehearsal. Old facts are re-dreamed from the weights, and by night 3 only 24% of those dreams are right, so most of what is rehearsed about corpus A is wrong.

4.6 Filtering rehearsal

A replay dream is wrong because the weights no longer hold the fact, so the filter has to detect that without the answer. We tested three signals, all computed from the model's own dreams (Table 8). *Agreement*: re-dream a fact only if at least K of its seven dreams agree. This made rehearsal worse at every K: the facts the model is most sure of are as often wrong as the others (22 to 50% right), and dropping the less sure ones removed the right answers along with the wrong ones; corpus A kept 18 to 27%. Majority voting is known to lock in confident errors under self-training [8]; we see

Table 6: Sparse writes on top of the filters (three seeds, same loop as Table 7).

	A kept		retained of 180		
seed	filters	+ sparse	filters	+ sparse	drift
0	44.1%	70.6%	∼70	∼76	+0.040
1	63.0%	92.3%	∼73	∼73	+0.046
2	45.0%	80.0%	∼59	∼73	+0.047
mean	50.7%	**81.0%**	∼67	∼**74**	+0.044

the same thing in rehearsal. *Ablation*: re-dream each old fact a second time with the lifetime tables switched off; if the vote is the same, the tables add nothing to that answer and it is the backbone's guess, so the fact is skipped. *Stability*: remember each fact's voted answer from its last dream (its day dream, or the previous night) and skip it when the answer has changed, in the spirit of the flip rate of Lin et al. [8]. Day dreams are never filtered, so new information can still overwrite old.

With both ablation and stability, rehearsal is right 100 and 79% of the time on nights 2 and 3, against 47 and 24% unfiltered, and the model ends with about 70 of 180 facts against 56, at the same drift. Corpus A keeps 44% against 50%: the filters remove the facts the weights already lost, so they protect what is still known and cannot restore what is not, and without the answer nothing can.

That single run understates the filters. Over three seeds, which change the tables' initialization, the order of each day's stream and which held-out corpus comes first (Table 7), the unfiltered loop keeps 39.8% of corpus A on average, so the 50.0% of round 7c was a lucky draw, and one seed collapses to 17.5% as its replay dreams fall to 14% correct. With both filters corpus A keeps 50.7% on average and wins in two of three seeds, and the model retains more facts in all three (67 against 54 of 180 on average). Both pre-registered criteria hold: the filters help keeping and retention.

4.7 Sparse writes

With rehearsal mostly right, what remains is forgetting during the sleep itself. Following sparse memory finetuning [9], we let each night change only the table rows its documents use far more than ordinary text: rows are ranked by their hit count in the night's dreams times the log inverse of their frequency on FineWeb, and only the top quarter of the rows the night hits may change, about 2,500 of 524,288 (0.5%). All other rows are restored after every step, which also stops weight decay from shrinking them. The shared projections still train. We predicted little gain, since Section 4.2 placed interference in the shared projections. That was wrong (Table 6): with the filters on, corpus A keeps 81.0% on average against 50.7%, in all three seeds, and the model retains 74 facts against 67, while learning is unchanged. Once rehearsal is clean, which rows a night overwrites matters. The cost is drift, which doubles to +0.044 on average, inside the budget of +0.05 but close to it.

4.8 Sleeping during the day

Can consolidation run continuously, as text scrolls out of the cache, instead of once a day? We split each day's sleep into six, one per 512-token window as it leaves the fast tier, with the same total number of steps. It fails. After the first small sleep, the earlier windows of the same day are already old and are re-dreamed from weights that have seen them for 2000 steps; those dreams are right 20% of the time at first and 0% by the end, and the model trains on its own errors all day (corpus A learned 16.7%, kept 0%). The filters stop the collapse but not the deficit: 30% learned and 22% kept, against 56.7% and 44% for one sleep per day. In this system the slow tier needs a

Table 7: Three seeds of the full loop on the round-7c model, without and with the ablation and stability filters. Night-1 learning is identical by construction (no old facts yet).

	A kept		retained of 180	
seed	none	filters	none	filters
0 (A learned 56.7%)	50.0%	44.1%	∼56	∼70
1 (A learned 45.0%)	51.8%	63.0%	∼68	∼73
2 (A learned 66.7%)	17.5%	45.0%	∼39	∼59
mean	39.8%	**50.7%**	∼54	∼**67**
replay right, n2/n3	46 / 27%	100 / 77%		

Table 8: Rehearsal filters on the round-7c model (same loop as Table 5, one seed). Replay accuracy is over the old facts that were rehearsed (nights 2 / 3); retained = facts of 180 recalled at the end. Micro-sleep learns corpus A at 16.7% (30.0% with both filters) against 56.7%.

filter	replay right	A kept	retained	drift
none (7c)	47 / 24%	50.0%	∼56	+0.024
agree ≥4/5/6 of 7	22–50%	18–27%		≤+0.025
ablation	50 / 39%	44.1%	∼67	+0.022
ablation + stability	**100 / 79%**	44.1%	∼**70**	+0.022
micro-sleep	20 → 0%	0%		+0.018
micro + both	45 / 28%	22.2%	∼13	+0.019

whole day's evidence before its own dreams can be trusted for rehearsal. Lee et al. [7] consolidate per context window successfully, but their sleep is a learned recurrence trained end to end, not rehearsal from self-labelled dreams.

5 Related work

Test-time memory. TTT layers [15] and Titans [2] learn a fast-weight memory during the forward pass; Nested Learning and Hope [1] add memories at several update frequencies. Gated DeltaNet [21] uses a delta-rule state with per-token gates. Our negative results for these mechanisms are specific to a 134M model, a sparse retrieval signal and cross-document state; they say the bottleneck was learning to address, not capacity. **Retrieval over cached states.** Transformer-XL [6] and Memorizing Transformers [20] attend to cached keys and values; MemoryLLM and M+ [18, 19] maintain a self-updated latent memory pool in an 8B model. **Sleep for language models.** Two concurrent works also give language models a sleep phase. Behrouz et al. [3] consolidate by distilling a smaller model's memories into a larger network (knowledge seeding) and then dream: the model uses reinforcement learning to generate a synthetic curriculum that rehearses and refines its knowledge. Lee et al. [7] run offline recurrent passes over the accumulated context and write it into the fast weights of SSM blocks through a learned local rule before clearing the KV cache. Ours differs in what is written and how it is labelled: dreams are completions of answer-free cues, the only label is a vote among the model's own dreams, the target is a sparse hashed table rather than network weights, and every write is gated against held-out loss and old recall with rollback. HOLA [5] pairs a delta-rule state with a bounded exact KV cache, the same split as our backbone plus fast tier. **Consolidation and replay.** Complementary learning systems [11]; pseudo-rehearsal [13]; generative replay [14]; sleep-like replay in networks [16]. SEAL [22] has a model write its

own finetuning data. Self-consistency [17] is our voting rule. A language model's own samples are effective replay against forgetting unless the model is near capacity [10]. Sparse memory finetuning [9] limits interference by updating only memory slots that new knowledge activates more than pretraining did; we adopt its row selection for the tables (Section 4.7).

6 Limitations

The final comparison uses three seeds; every other arm is one seed, and several effects sit within one to two standard errors. Corpora of 60 facts make each corpus-level percentage a handful of facts. The answer vocabulary has ten names. The facts are synthetic and the paraphrase probes are light. The fast tier is a bounded cache, so facts older than one window rely on sleep. The XL model is worse at ordinary language modeling than the matrix-memory model (3.51 against 3.34), and a cache bias did not close the gap. Cue tuning trains the exact cue format the night uses; held-out subjects and per-sample answers rule out memorized facts, but not that the read is specific to this format. Rehearsal can only protect facts the weights still hold, and consolidation within the day (Section 4.8) fails. Scale is 134M parameters and 393M tokens per round; paraphrase-invariant retrieval at this size is capacity-limited (30% in-context ceiling on our hardest probe level). Lifetime tables are trained by gradient descent; one-shot writes into the fast-weight memory across documents failed (0%).

7 Conclusion

A small model can learn by day and consolidate by night with no ground truth after the day itself: it dreams about what it read, votes on its dreams, sleeps on them, and keeps the result. The stage that failed first was not storage but the step from memory to a dream: the cache held the facts and the model could not answer a cue from it until it was trained to. After that it learns more than half of a day and, with filtered rehearsal and sparse writes, keeps four fifths of that. Rehearsal from its own dreams was mostly wrong until two label-free checks, whether the tables change the answer and whether the answer is stable across nights, made it mostly right, and confining each night's write to the rows its dreams use took keeping from half to four fifths; what it costs now is val drift close to the budget. The two lessons we would pass on are methodological. Measure the backbone's own ceiling on every probe before setting a bar. And when a new memory module stores information but does not use it, check whether the model already has a retrieval circuit you could hand the memory to.

References

[1] Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. Nested learning: The illusion of deep learning architectures. In *Advances in Neural Information Processing Systems*, 2025. arXiv:2512.24695.

[2] Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*, 2025.

[3] Ali Behrouz, Farnoosh Hashemi, Adel Javanmard, and Vahab Mirrokni. Language models need sleep: Learning to self-modify and consolidate memories. *arXiv preprint arXiv:2606.03979*, 2026.

[4] Xin Cheng, Rui Tian, Wangding Zeng, Damai Dai, et al. Conditional memory via scalable lookup: A new axis of sparsity for large language models. *arXiv preprint arXiv:2601.07372*, 2026.

[5] Wanyun Cui. A hippocampus for linear attention: An exact memory for what the recurrent state forgets. *arXiv preprint arXiv:2607.02303*, 2026.

[6] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of ACL*, 2019.

[7] Sangyun Lee, Sean McLeish, Tom Goldstein, and Giulia Fanti. Do language models need sleep? offline recurrence for improved online inference. *arXiv preprint arXiv:2605.26099*, 2026.

[8] Hongxiang Lin, Zhirui Kuai, Erpeng Xue, and Lei Wang. Detecting and mitigating the correct-answer extinction window in test-time reinforcement learning with majority voting. *arXiv preprint arXiv:2605.19444*, 2026.

[9] Jessy Lin, Luke Zettlemoyer, Gargi Ghosh, Wen-Tau Yih, Aram Markosyan, Vincent-Pierre Berges, and Barlas Oğuz. Continual learning via sparse memory finetuning. *arXiv preprint arXiv:2510.15103*, 2025.

[10] Martin Marek, Dongkyu Cho, Shikai Qiu, Rumi Chunara, Pavel Izmailov, and Andrew Gordon Wilson. Forgetting in language models: Capacity, optimization, and self-generated replay. *arXiv preprint arXiv:2605.26097*, 2026.

[11] James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3): 419–457, 1995.

[12] Guilherme Penedo, Hynek Kydlíček, et al. The FineWeb datasets: Decanting the web for the finest text data at scale. In *NeurIPS Datasets and Benchmarks Track*, 2024.

[13] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

[14] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, 2017.

[15] Yu Sun, Xinhao Li, Karan Dalal, et al. Learning to (learn at test time): RNNs with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

[16] Timothy Tadros, Giri P. Krishnan, Ramyaa Ramyaa, and Maxim Bazhenov. Sleep-like unsupervised replay reduces catastrophic forgetting in artificial neural networks. *Nature Communications*, 13, 2022.

[17] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

[18] Yu Wang et al. MEMORYLLM: Towards self-updatable large language models. In *International Conference on Machine Learning*, 2024.

[19] Yu Wang et al. M+: Extending MemoryLLM with scalable long-term memory. In *International Conference on Machine Learning*, 2025. arXiv:2502.00592.

[20] Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *International Conference on Learning Representations*, 2022.

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

[22] Adam Zweiger, Jyothish Pari, Han Guo, Ekin Akyürek, Yoon Kim, and Pulkit Agrawal. Self-adapting language models. *arXiv preprint arXiv:2506.10943*, 2025.