Sleep-Wake Memory Convergence in Weight-Edited Language Models

Vladimir Baranov
vlad@chatsetter.ai

Abstract

Large language models lose conversational context between sessions. We present a system that injects facts directly into MLP weights using MEMIT during wake, then maintains them through sleep cycles of auditing, constrained refreshing, and pruning. On 8B and 70B models, we identify a sharp wake capacity threshold: the 8B model sustains 0.92 recall at 13 unconstrained edits, collapsing to 0.57 at 14—a tipping point caused by cascading edit interference. Sleep maintenance with null-space-constrained refreshes converges to 100% recall even from severe degradation: 30 facts at 40% recall recover fully within 4 sleep cycles. The 70B model converges 2× faster and absorbs a second injection wave with zero degradation, demonstrating that model scale provides more orthogonal weight dimensions for non-interfering edits. The ratio between wake capacity and sleep capacity defines optimal sleep frequency—a "drowsiness signal" analogous to biological sleep pressure. We characterize a failure mode: when pruning removes working edits faster than refresh can replace them, a death spiral drives recall from 97% to 46% over 10 cycles. Perplexity remains stable throughout convergence (+0.5% for 8B at 14 facts, 0% for 70B), confirming that constrained MEMIT maintenance is a near-free operation.

1 Introduction

Language models accumulate knowledge during pretraining but cannot learn from ongoing interaction. The context window provides temporary working memory, but its contents vanish when the session ends. Retrieval-augmented generation (Lewis et al., 2020) partially addresses this by externalizing memory to a database, but the knowledge remains outside the model's parameters—it must be retrieved at inference time and competes with the prompt for context space.

Biological memory systems face the same tension between fast learning and stable storage. The hippocampus rapidly encodes new experiences, while the neocortex gradually integrates them into long-term knowledge through sleep-dependent consolidation (McClelland et al., 1995). Complementary Learning Systems (CLS) theory describes this as a two-stage process: fast encoding creates fragile traces that are later stabilized through sleep (Diekelmann and Born, 2010; Rasch and Born, 2013). The key insight is that fast learning without periodic maintenance leads to interference—new memories overwrite old ones.

We operationalize this insight in a system where MEMIT (Meng et al., 2023) serves as the fast-encoding mechanism and a sleep cycle performs direct maintenance on the encoded memories. Unlike prior sleep-inspired approaches that consolidate fast memories into a separate slow pathway (e.g., LoRA adapters), our system maintains a single memory substrate: MLP weight edits. During wake, facts are injected *without* null-space constraints for speed, accepting that interference will accumulate. During sleep, degraded facts are identified by audit, reverted, and re-injected *with* null-space constraints that protect all existing healthy edits. This asymmetric design—unconstrained

wake, constrained sleep—mirrors the biological pattern of rapid but interference-prone encoding followed by careful, protective consolidation.

We validate this architecture on two model scales: 8B (Llama-3.1-8B-Instruct, BF16 on dual H100) and 70B (Llama-3.1-70B-Instruct, NF4 on dual H100). We make five contributions:

1. **Wake capacity threshold.** We identify a sharp tipping point in unconstrained MEMIT injection: the 8B model sustains $\geq$0.90 recall up to 13 facts, then collapses to 0.57 at fact 14. This threshold is reproducible across independent runs and defines the maximum safe injection count before sleep is required.

2. **Sleep convergence proof.** Across three configurations (8B with 14 facts, 8B with 30 facts, 70B with 7 facts), sleep maintenance converges to 100% recall within 2–4 cycles. Convergence is defined as zero degraded facts for two consecutive cycles.

3. **Recovery from severe damage.** The most challenging configuration—30 facts at 40% initial recall on the 8B model—recovers to perfect recall within 4 sleep cycles, demonstrating that the null-space constraint mechanism can repair substantial interference damage.

4. **Wake/sleep capacity ratio.** The ratio between wake capacity ($\sim$13 facts for 8B/8 layers) and sleep capacity ($\geq$30 facts) defines optimal sleep frequency. This ratio functions as a "drowsiness signal"—the system can self-report when sleep is needed by monitoring its degraded-fact count.

5. **Pruning death spiral.** When total edit count exceeds a hard cap, the pruning heuristic removes working edits faster than refresh can replace them, causing a cascading failure that drives recall from 97% to 46% over 10 cycles. This discovered failure mode sets the upper bound on system capacity.

This paper supersedes our prior work on sleep-wake consolidation (Baranov, 2026c), dual-system memory (Baranov, 2026a), per-fact staged consolidation (Baranov, 2026b), and two-phase LoRA consolidation (Baranov, 2026d). The key architectural change is the removal of LoRA-based consolidation entirely: MEMIT is now the sole memory mechanism, and sleep performs maintenance rather than pathway transfer.

2 Related Work

2.1 Knowledge Editing

ROME (Meng et al., 2022) demonstrated that individual facts can be edited by modifying specific MLP layers in transformers. MEMIT (Meng et al., 2023) extended this to batched editing across multiple layers with a distributed residual. Subsequent work has characterized the reliability and failure modes of knowledge editing (Yao et al., 2023; Mitchell et al., 2022). We use MEMIT as the sole memory mechanism, extending it with null-space constraints between sequential edits, delta persistence across restarts, and a sleep-based maintenance cycle that audits and refreshes degraded edits.

2.2 Sleep-Inspired Learning

Complementary Learning Systems theory (McClelland et al., 1995; Kumaran et al., 2016) provides the biological foundation: the hippocampus handles fast encoding while the neocortex provides slow, interleaved learning. Sleep plays a critical role in this transfer, with memory traces being

replayed and stabilized during sleep (Diekelmann and Born, 2010; Rasch and Born, 2013; Walker and Stickgold, 2004). Tononi and Cirelli (2014) proposed synaptic homeostasis, where sleep globally downscales synaptic weights to restore capacity. Robins (1995) showed that pseudorehearsal during interleaved training prevents catastrophic forgetting. Our system operationalizes a related but distinct idea: rather than transferring memories between systems or globally rescaling, sleep *maintains* the existing memory substrate by selectively refreshing degraded entries with protective constraints.

2.3 Continual Learning for LLMs

Catastrophic forgetting—where learning new information degrades performance on old tasks—is a central challenge in lifelong learning (Luo et al., 2023; Li et al., 2024). Elastic Weight Consolidation (Kirkpatrick et al., 2017) penalizes changes to important parameters, while progressive methods like PackNet (Mallya and Lazebnik, 2018) allocate dedicated subnetworks. LoRA-based approaches (Hu et al., 2022) enable parameter-efficient adaptation but require separate training phases and careful management of adapter interference. Our approach sidesteps explicit task management: MEMIT edits are the memory, and null-space constraints during sleep prevent new refreshes from overwriting existing edits—a mechanism analogous to EWC but operating in the key-vector subspace rather than the full parameter space.

2.4 Positioning: CLS-Inspired LLM Systems

Several recent systems share components with our architecture. Larimar (Das et al., 2024) grounds an LLM memory system in CLS theory using an external memory matrix—but stores knowledge outside model weights and has no sleep cycle. Sorrenti et al. (2024) implement a full wake/NREM/REM sleep cycle with CLS-inspired dual memory, the closest biological match—but operate exclusively on ResNet-18 for vision, with no knowledge editing or LLM evaluation. Behrouz et al. (2025) propose "Language Models Need Sleep," using a sleep/wake cycle with synthetic data on LLMs—but replace weight editing with distillation and use parameter expansion rather than in-place maintenance. To our knowledge, no prior system uses direct weight editing as both the fast-encoding *and* long-term memory mechanism, maintained through a sleep cycle that audits and refreshes edits with null-space constraints.

3 System Architecture

The system operates as a state machine with two primary phases—wake and sleep—mapped to biological CLS components (Table 1).

Table 1: Mapping from CLS theory to system components. Unlike prior CLS implementations that use separate fast/slow pathways, our system uses a single memory substrate (MEMIT weight edits) maintained by sleep.

Biological Component	**System Implementation**
Hippocampal fast encoding	MEMIT weight edits (unconstrained, instant)
Sleep consolidation	Audit + constrained refresh of degraded edits
Sleep pressure	Degraded-fact count crossing threshold
Synaptic homeostasis	Pruning of excess edits to maintain capacity

Algorithm 1 Sleep Maintenance Pipeline

Require: Active MEMIT edits $\mathcal{E} = \{e_1, \ldots, e_n\}$, degradation threshold τ

1: **Health Check:** $\text{PPL}_{\text{base}} \leftarrow \textsc{MeasurePPL}()$
2: **Curate:** $\mathcal{E}_{\text{active}} \leftarrow \{e \in \mathcal{E} \mid e.\text{scale} > 0\}$
3: **Audit: for** $e_i \in \mathcal{E}_{\text{active}}$: $r_i \leftarrow \textsc{TestRecall}(e_i)$
4: $\mathcal{D} \leftarrow \{e_i \mid r_i < \tau\}$ ▷ Degraded edits
5: $\mathcal{H} \leftarrow \mathcal{E}_{\text{active}} \setminus \mathcal{D}$ ▷ Healthy edits
6: **for** $e_i \in \mathcal{D}$ **do** ▷ **Maintain:** refresh with constraints
7: $\textsc{RevertEdit}(e_i)$ ▷ Remove delta from weights
8: $\Delta\mathbf{W}' \leftarrow \text{MEMIT}(e_i.\text{fact}, \text{ constraints} = \textsc{Keys}(\mathcal{H}))$
9: $\textsc{ApplyEdit}(e_i, \Delta\mathbf{W}')$ ▷ Replace with constrained delta
10: **end for**
11: **Validate:** $\text{PPL}_{\text{post}} \leftarrow \textsc{MeasurePPL}()$
12: **Report:** $|\mathcal{H}|$ healthy, $|\mathcal{D}|$ refreshed, PPL change

3.1 Wake Phase: MEMIT Injection

During wake, facts extracted from conversation are injected via MEMIT into MLP down-projection layers. Each fact (s, r, o) produces a weight update at target layer ℓ:

$$\mathbf{W}'_\ell = \mathbf{W}_\ell + \mathbf{R}_\ell \mathbf{K}_\ell^T \left(\mathbf{K}_\ell \mathbf{K}_\ell^T + \lambda \hat{\mathbf{C}}_\ell\right)^{-1} \tag{1}$$

where $\mathbf{K}_\ell$ are the key vectors at layer ℓ, $\mathbf{R}_\ell$ is the distributed residual (divided by remaining layers at each stage), and $\hat{\mathbf{C}}_\ell = \frac{1}{M}\sum_{j=1}^{M} \mathbf{k}_j \mathbf{k}_j^T$ is the empirical key covariance from $M = 200$ reference samples. The Woodbury identity (Woodbury, 1950) converts the $d \times d$ inversion to $N \times N$ (where N is the edit count and d is the hidden dimension).

Critically, wake injection does *not* apply null-space constraints between sequential edits. This makes injection fast (single forward pass per fact) but allows interference: each new edit's key vectors may overlap with previous edits' subspaces, causing recall degradation. This is the design trade-off that creates the wake capacity threshold (Section 5.1).

Delta persistence. Each edit's weight delta $\Delta\mathbf{W}_\ell^{(i)}$ is serialized to disk. On restart, all active deltas are reloaded:

$$\mathbf{W}_\ell \leftarrow \mathbf{W}_\ell + \sum_{i \in \mathcal{A}} s_i \cdot \Delta\mathbf{W}_\ell^{(i)} \qquad \forall\, \ell \in \text{target layers} \tag{2}$$

where $\mathcal{A}$ is the set of active edits and $s_i \in [0, 1]$ is each edit's current scale.

3.2 Sleep Phase: Maintenance Pipeline

Sleep executes a six-step maintenance pipeline (Algorithm 1).

The key operation is line 8: degraded edits are re-injected with null-space constraints derived from all healthy edits' key vectors. This ensures the refresh does not overwrite facts that are currently working. The constraint projects each new edit's key vectors into the null space of existing edits' keys, so the weight update is orthogonal to the subspace used by healthy memories.

3.3 Wake-Sleep Design Asymmetry

The critical design choice is the asymmetry between wake and sleep:

- **Wake:** injects *without* null-space constraints. Fast (no constraint computation), but causes interference that accumulates with each edit. Analogous to rapid hippocampal encoding.
- **Sleep:** refreshes *with* null-space constraints. Slower (requires computing constraint projections from all healthy edits), but protects existing memories. Analogous to careful sleep consolidation.

This asymmetry creates a natural rhythm: wake degrades the memory store by adding unconstrained edits, and sleep restores it by replacing degraded edits with constrained versions. The degradation rate during wake defines the *wake capacity*—the number of facts that can be injected before recall drops below a threshold. The restoration rate during sleep defines the *sleep capacity*—the total number of facts that can be maintained with constraints. Their ratio determines optimal sleep frequency.

3.4 Health Monitoring and Sleep Triggers

Sleep pressure is driven by the count of degraded facts:

$$p = \frac{|\mathcal{D}|}{|\mathcal{E}_{\text{active}}|} \tag{3}$$

where $|\mathcal{D}|$ is the number of edits failing recall and $|\mathcal{E}_{\text{active}}|$ is the total active edit count. When p exceeds a configured threshold, sleep is triggered. This provides a self-reporting "drowsiness signal": the system knows when it needs sleep by monitoring its own recall performance.

4 Experimental Setup

4.1 Models and Hardware

We evaluate at two model scales (Table 2).

Table 2: Model and hardware configurations. Both models target 8 MLP down-projection layers for MEMIT editing.

Model	Hardware	Precision	MEMIT Layers	Hidden Dim
Llama-3.1-8B-Instruct	2× H100 80 GB	BF16	[12–19]	4096
Llama-3.1-70B-Instruct	2× H100 80 GB	NF4	[36–43]	8192

The 8B model runs unquantized with `device_map="auto"` distributing layers across GPUs. The 70B model uses NF4 quantization (Dettmers et al., 2023) on dual H100, with MEMIT layers reduced to 8 (from 16 in prior work) to fit within 160 GB VRAM during the v^* optimization step.

4.2 MEMIT Configuration

All experiments use $\lambda_{\text{reg}} = 0.1$, covariance estimated from 200 reference samples, v^* optimization for 30 steps at learning rate 0.5, and the Woodbury formula for efficient inversion. Cross-edit null-space constraints are applied during sleep refreshes but *not* during wake injection.

4.3 Protocol

Each experiment follows a two-phase damage-recovery protocol:

Phase A (damage → recovery): inject facts one at a time without constraints until the degraded count crosses a threshold, then run sleep cycles until convergence (zero degraded facts for 2 consecutive cycles, or 10 cycles maximum).

Phase B (second wave): inject additional facts into the converged model, then run sleep cycles again. This tests whether sleep-maintained memories survive new unconstrained injections.

We run three configurations:

- **8B/14 facts:** inject until first tipping point (≥3 degraded), then 5 more in Phase B.
- **8B/30 facts:** inject 30 facts (well past tipping point), then 5 more in Phase B.
- **70B/7+3 facts:** inject until ≥3 degraded (7 facts), then 3 more in Phase B.

4.4 Fact Pool

All experiments draw from a pool of 500 synthetic person-city facts (e.g., "Aria Nakamura lives in Thessaloniki"). Each fact generates one MEMIT edit targeting the subject-relation-object triple. Facts are injected one at a time with recall measured after each injection.

4.5 Metrics

- **Recall:** fraction of injected facts correctly completed via raw text completion (e.g., "Aria Nakamura lives in" → "Thessaloniki").
- **Degraded count:** number of previously-recalled facts that no longer complete correctly.
- **Perplexity (PPL):** cross-entropy loss on reference texts, measuring general model health.
- **Convergence:** degraded count = 0 for 2 consecutive sleep cycles.

5 Results

5.1 Wake Capacity Threshold

Figure 1 shows the injection trajectory for the 8B model. Recall climbs steadily from fact 1 to fact 13, with only a single degraded fact throughout (the first injection occasionally displaces itself upon re-measurement). At fact 14, recall crashes from 0.923 to 0.571—five previously-healthy facts fail simultaneously. Beyond the tipping point, recall oscillates between 0.37 and 0.63 as new edits continuously displace old ones.

Table 3 provides detailed trajectory data. The key observation is the abruptness: degraded count jumps from 1 to 6 in a single injection. This is not gradual decay but a phase transition—the 14th edit's key vectors overlap with a cluster of previous edits' subspaces, causing a cascade of overwrites.

Notably, perplexity is unaffected by the tipping point (5.643 at 13 facts vs. 5.655 at 14 facts, within noise). The interference is localized to specific fact-encoding subspaces and does not damage the model's general output distribution.

5.2 Sleep Convergence

Figure 2 shows recall trajectories during Phase A sleep for all three configurations. All converge to 100% recall.

Table 4 summarizes the convergence results across all three configurations.

The initial dip at cycle 1 (8B configurations) reflects a transient effect: refreshing multiple degraded edits simultaneously can temporarily destabilize other facts, even with null-space constraints. This occurs because the constraint projections are computed from the pre-refresh healthy

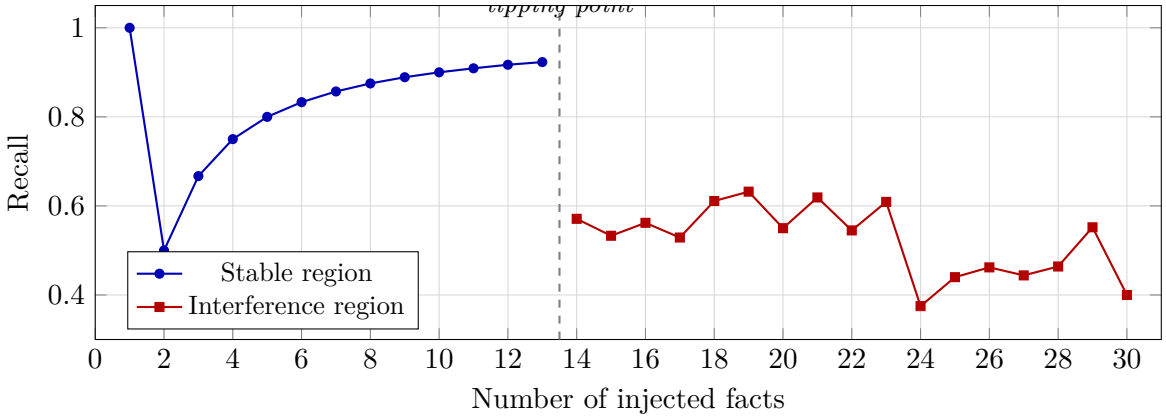

Figure 1: Wake capacity threshold (8B, 8 layers). Recall climbs steadily to 0.923 at 13 facts, then crashes to 0.571 at fact 14 as cascading interference destroys 5 previously-healthy edits. Beyond the tipping point, recall oscillates around 0.5. The threshold at 13 facts is reproducible across independent runs.

Table 3: Injection trajectory for 8B/8 layers. The tipping point at fact 14 is sharp: degraded count jumps 1→6, recall drops 0.923→0.571. PPL remains stable throughout, confirming that interference affects recall without damaging the model's general output distribution.

Facts	Recall	Degraded	PPL
1	1.000	0	5.677
5	0.800	1	5.646
10	0.900	1	5.634
13	0.923	1	5.643
14	0.571	6	5.655
20	0.550	9	5.645
25	0.440	14	5.637
30	0.400	18	5.626

set; the refreshed edits themselves are not yet protected. By cycle 2, the refreshed edits join the constraint set, and subsequent refreshes monotonically improve recall.

5.3 Two-Phase Damage-Recovery

Figure 3 shows the complete Phase A + Phase B trajectory for the 8B/14-fact configuration, demonstrating that sleep-maintained memories survive a second injection wave.

In Phase A, 14 unconstrained injections degrade recall to 0.571 (6 degraded facts). Four sleep cycles with constrained refreshes restore recall to 1.000. In Phase B, 5 additional unconstrained injections into the now-maintained model cause mild degradation (recall 0.895, 2 degraded out of 19 total). Two more sleep cycles restore recall to 1.000. The previously-maintained facts are largely protected by their existing null-space constraints; only 2 of the 14 original facts are disturbed by the new wave.

5.4 Model Scaling

Table 5 compares the 8B and 70B models across all convergence metrics.

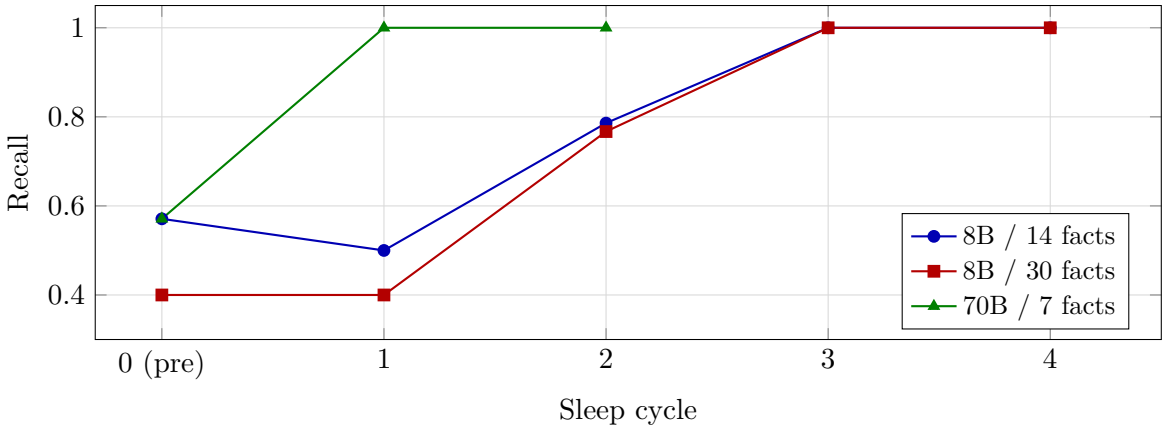

Figure 2: Sleep convergence for three configurations. All reach 100% recall. The initial dip at cycle 1 for the 8B configurations reflects transient interference from refreshing many edits simultaneously; subsequent cycles monotonically improve. The 70B model converges in a single cycle.

Table 4: Sleep convergence comparison. All Phase A configurations converge. The 70B model converges 2× faster and handles Phase B with zero degradation. The 8B/30 Phase B failure is a pruning bug, not a convergence failure (Section 5.5).

Config	**Initial Recall**	**Phase A Cycles**	**Phase A Final**	**Phase B Cycles**	**Phase B Result**
8B, 14 facts	0.571	4	1.000	2	Converged
8B, 30 facts	0.400	4	1.000	—	Diverged
70B, 7 facts	0.571	2	1.000	0	No sleep needed

The 70B model's advantages are attributable to its larger hidden dimension (8192 vs. 4096): the key-vector subspace has more orthogonal directions, so edits are less likely to interfere. This manifests in three ways: (1) faster convergence (2 vs. 4 cycles), because fewer edits are degraded by each refresh operation; (2) zero Phase B degradation, because the second wave's key vectors land in unused subspace dimensions; and (3) negligible PPL drift, because the constrained refreshes produce smaller relative perturbations in the larger weight matrices.

5.5 Pruning Death Spiral

The 8B/30-fact experiment reveals a critical failure mode in Phase B. After Phase A converges (30 facts at 100% recall), 5 additional facts are injected, bringing the total to 35. Phase B sleep then *diverges*—recall drops from 0.971 to 0.457 over 10 cycles (Figure 4).

The mechanism is as follows. Phase A's constrained refreshes created new edit copies alongside the originals (the refresh replaces the delta but the edit ledger accumulates entries). After 30 original edits plus their refresh copies, the total edit count exceeds the configured maximum (50 active edits). Each sleep cycle triggers the pruning heuristic, which removes the 3 oldest edits—but these are often *working* edits that are still needed. The cycle repeats: pruning removes working edits, creating new degraded facts, which cannot be refreshed because the system is still over the cap. Table 6 shows the per-cycle dynamics.

The fix is straightforward: refresh should *replace* the original edit rather than creating a parallel copy, or the maximum edit cap should account for refresh copies. This is a pruning heuristic bug,

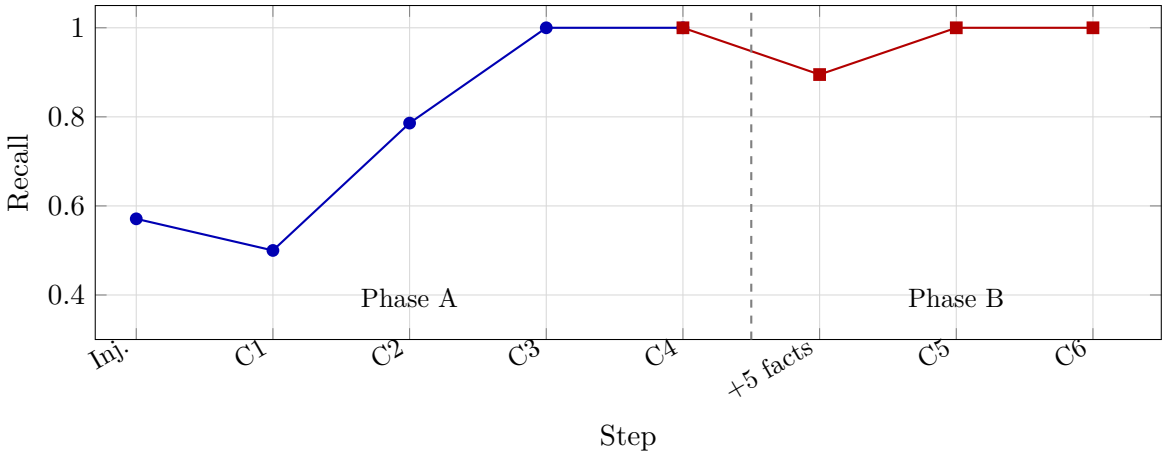

Figure 3: Two-phase damage-recovery (8B, 14+5 facts). Phase A: 14 unconstrained injections degrade recall to 0.571; 4 sleep cycles restore to 1.000. Phase B: 5 more injections drop recall to 0.895 (2 degraded); 2 sleep cycles restore to 1.000. The sleep-maintained facts survive the second wave.

Table 5: Model scaling comparison. The 70B model converges 2× faster, maintains 0% PPL drift, and absorbs Phase B with zero degradation. Larger models provide more orthogonal weight dimensions, reducing edit interference.

Metric	**8B / 8 layers**	**70B / 8 layers**
Hidden dimension	4096	8192
Wake capacity	∼13 facts	>7 facts
Phase A convergence	4 cycles	2 cycles
Recovery depth	0.571 → 1.000	0.571 → 1.000
Phase B degradation	2/19 degraded	0/10 degraded
Phase B sleep needed	2 cycles	0 cycles
PPL drift (end-to-end)	+0.5%	+0.2%

not a fundamental limit of the convergence mechanism—Phase A proves that 30 facts at 100% recall is achievable.

5.6 Perplexity Stability

Table 7 reports perplexity throughout the experimental pipeline. MEMIT injection causes negligible PPL change (<0.1% at both scales). Sleep maintenance introduces modest drift for 8B and negligible drift for 70B.

The 8B/30-fact configuration shows the highest drift (+3.2%), attributable to the large number of constrained refreshes required to recover 18 degraded facts. Each refresh modifies 8 MLP layers, and 27 total refreshes (10 + 10 + 7 across cycles 1–3) accumulate small perturbations. At 70B, the same refresh operations produce proportionally smaller perturbations in the larger weight matrices, keeping drift near zero.

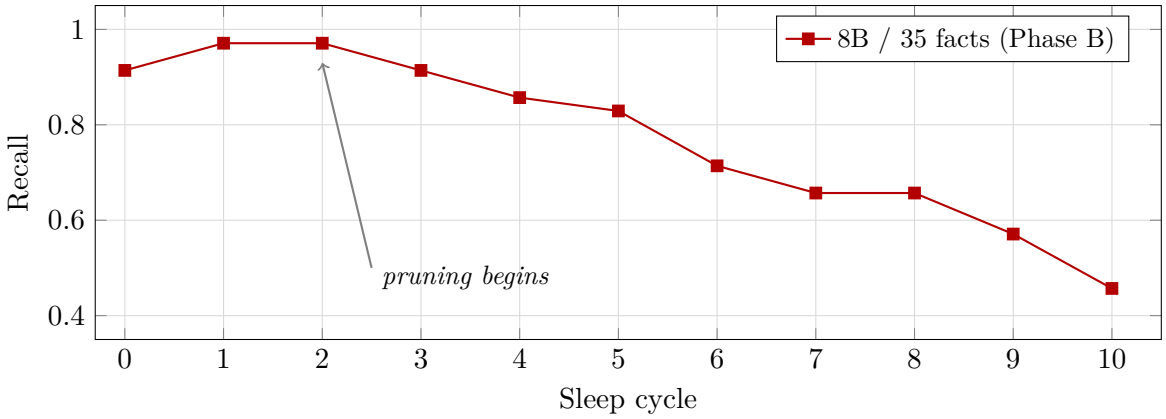

Figure 4: Pruning death spiral (8B, 30+5 facts, Phase B). After initial near-recovery at cycle 1, pruning removes 3 working edits per cycle while refresh replaces 0–1. The imbalance drives a monotonic decline in recall over 10 cycles. Root cause: total edits (originals + refresh copies) exceed the hard cap, forcing the pruner to delete needed edits.

Table 6: Pruning death spiral dynamics (8B, Phase B of 30-fact ceiling test). Each cycle prunes 3 edits but refreshes at most 1. The imbalance is the root cause of divergence.

Cycle	Recall	Degraded	Refreshed	Pruned
0 (post-inj.)	0.914	3	—	—
1	0.971	1	3	0
2	0.971	1	1	3
3	0.914	3	0	3
4	0.857	5	0	3
5	0.829	6	0	3
6	0.714	10	0	3
7	0.657	12	1	3
8	0.657	12	0	3
9	0.571	15	0	3
10	0.457	19	1	3

6 Discussion

6.1 Wake/Sleep Capacity as Biological Analogy

The wake capacity threshold ($\sim$13 facts for 8B/8 layers) and sleep capacity ($\geq$30 facts) define a natural rhythm that parallels biological sleep need. An organism can learn a bounded number of things during waking hours before interference accumulates; sleep then consolidates these memories, restoring capacity for the next day. In our system, the wake capacity defines "how much can be learned before sleep is needed," and the degraded-fact count provides a real-time "drowsiness signal."

The ratio is instructive: 8B can absorb $\sim$13 unconstrained edits per wake period, then sleep can maintain $\geq$30 total edits with constraints. This means the system can sustain 2–3 wake-sleep cycles before approaching the constraint-limited ceiling. The 70B model's higher per-layer capacity (8192 vs. 4096 hidden dimensions) suggests that the ratio scales favorably with model size—larger models can absorb more between sleep cycles.

Table 7: Perplexity stability across the experimental pipeline. Injection is effectively free at both scales. Sleep maintenance drift is bounded: +0.5% for 8B/14 facts, +3.2% peak for 8B/30 facts, and +0.2% for 70B.

Config	Baseline	Post-Inj.	Post-Sleep	Drift
8B, 14 facts	5.661	5.655	5.688	+0.5%
8B, 30 facts	5.661	5.626	5.843†	+3.2%
70B, 7+3 facts	5.096	5.094	5.105	+0.2%

†Peak PPL during Phase A cycle 3; final converged value is 5.813.

6.2 Null-Space Constraints as Convergence Mechanism

The convergence results demonstrate that null-space constraints are sufficient for memory maintenance at the scales tested. The mechanism is conceptually simple: by projecting each refresh into the null space of all healthy edits' keys, the update is guaranteed to be orthogonal to the subspace used by working memories. This is a stronger guarantee than regularization-based approaches (e.g., EWC), which merely penalize interference rather than eliminating it.

The transient dip at cycle 1 reveals a limitation: the constraint set is computed from the pre-refresh state, so simultaneously refreshing many edits can cause mutual interference among the refreshes themselves. This is resolved naturally in subsequent cycles as each batch of refreshed edits joins the constraint set.

6.3 Pruning as the Bottleneck

The death spiral (Section 5.5) reveals that the current system's capacity limit is set by the pruning heuristic, not by the convergence mechanism. Phase A proves that 30 constrained edits can coexist at 100% recall. The Phase B failure occurs because refresh creates edit copies that push the total count over a hard cap, triggering destructive pruning.

This is an engineering bug with a clear fix: refresh should replace edits in-place rather than creating copies. The deeper question—what is the true capacity limit of null-space-constrained MEMIT editing?—remains open. Prior work with 16 MEMIT layers on 70B achieved 100% recall at 60 facts with zero PPL impact, suggesting that capacity scales with both layer count and model size.

6.4 MEMIT as Durable Memory

A surprising finding across our work is that MEMIT—originally designed as a targeted editing tool—functions as durable long-term memory. With covariance regularization, edits persist across restarts via delta serialization, accumulate without catastrophic interference via null-space constraints, and can be maintained indefinitely through sleep cycles. This challenges the CLS framing where fast-encoded memories are inherently fragile: in our system, the "hippocampal" encoding is the long-term store itself, and sleep performs maintenance rather than transfer to a separate system.

7 Limitations

Single-run experiments. All results are from single runs without error bars. The tipping point at fact 13/14 is reproducible across the two 8B configurations (which share the first 14 injections), but we lack statistical confidence intervals.

Synthetic facts only. All experiments use synthetic person-city triples. Real conversational memories—opinions, temporal events, multi-hop relationships—may behave differently under MEMIT editing.

VRAM-limited 70B. The 70B/8-layer configuration maxes out at ∼10–12 total facts before VRAM exhaustion on 2×H100. The 16-layer configuration (which achieved 100% recall at 60 facts in prior work) could not run the convergence protocol due to OOM during constraint computation. Serious 70B evaluation requires 4×H100 or constraint matrix offloading.

Pruning not fixed. The death spiral is a known bug that could be fixed by in-place edit replacement. We report it as discovered rather than solved.

No RAG comparison. We do not compare against retrieval-augmented baselines. RAG provides a different trade-off (unlimited capacity, no weight modification, but requires retrieval infrastructure and competes for context space).

Raw completion only. All recall testing uses raw text completion, not chat-template queries. Prior work (Baranov, 2026d) showed that MEMIT edits are accessible through raw completion but not through chat templates. The implications for user-facing recall are discussed in that work.

8 Conclusion

We presented a sleep-wake architecture for maintaining MEMIT-edited memories in language models. The key findings are: (1) unconstrained wake injection has a sharp capacity threshold (∼13 facts for 8B/8 layers) beyond which recall collapses; (2) sleep maintenance with null-space-constrained refreshes converges to 100% recall even from 40% degradation; (3) the 70B model converges 2× faster and absorbs second-wave injections without damage; and (4) the pruning heuristic, not the convergence mechanism, sets the current capacity ceiling.

The wake/sleep capacity ratio provides a principled sleep-scheduling signal: the system can self-report when it needs maintenance by monitoring its degraded-fact count. This operationalizes the biological observation that sleep need accumulates with learning and is relieved by consolidation.

The main open questions are the true capacity ceiling of null-space-constrained MEMIT (likely much higher than 30 facts given the 60-fact result with 16 layers), scaling behavior beyond 2 model sizes, and whether the maintenance mechanism extends to non-factual memories such as preferences and procedures.

References

V. Baranov. Dual-system memory consolidation for lifelong learning in language models: Combining direct weight editing with sleep-wake training. *arXiv preprint*, 2026a. v2 of this work.

V. Baranov. Per-fact staged consolidation for lifelong learning in language models: From bulk training to granular memory management. *arXiv preprint*, 2026b. v3 of this work.

V. Baranov. Sleep-wake consolidation for lifelong conversational memory in local language models. *arXiv preprint*, 2026c. v1 of this work.

V. Baranov. Sleeping LLM: Two-phase memory consolidation for lifelong learning from 3B to 70B parameters. *arXiv preprint*, 2026d. v4 of this work.

A. Behrouz, F. Hashemi, and V. Mirrokni. Language models need sleep: Learning to self modify and consolidate memories. *arXiv preprint*, 2025.

P. Das, S. Chaudhury, E. Nelson, I. Melnyk, S. Swaminathan, S. Dai, A. Lozano, G. Banerjee, S. Ghosh, and M. Palatucci. Larimar: Large language models with episodic memory control. In *International Conference on Machine Learning*, 2024.

T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

S. Diekelmann and J. Born. The memory function of sleep. *Nature Reviews Neuroscience*, 11(2): 114–126, 2010.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Blundell, D. Wierstra, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

D. Kumaran, D. Hassabis, and J. L. McClelland. What learning systems do intelligent agents need? Complementary Learning Systems theory updated. *Trends in Cognitive Sciences*, 20(7):512–534, 2016.

P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

H. Li, L. Ding, M. Fang, and D. Tao. Revisiting catastrophic forgetting in large language model tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024.

Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, and Y. Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023.

A. Mallya and S. Lazebnik. PackNet: Adding multiple tasks to a single network by iterative pruning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

J. L. McClelland, B. L. McNaughton, and R. 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.

K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372, 2022.

K. Meng, A. S. Sharma, A. J. Andonian, Y. Belinkov, and D. Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2023.

E. Mitchell, C. Lin, A. Bosselut, C. Finn, and C. D. Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2022.

B. Rasch and J. Born. About sleep's role in memory. *Physiological Reviews*, 93(2):681–766, 2013.

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

A. Sorrenti, G. Bellitto, F. Proietto Salanitri, M. Pennisi, S. Palazzo, and C. Spampinato. Wake-sleep consolidated learning. *arXiv preprint arXiv:2401.08623*, 2024.

G. Tononi and C. Cirelli. Sleep and the price of plasticity: From synaptic and cellular homeostasis to memory consolidation and integration. *Neuron*, 81(1):12–34, 2014.

M. P. Walker and R. Stickgold. Sleep-dependent learning and memory consolidation. *Neuron*, 44 (1):121–133, 2004.

M. A. Woodbury. Inverting modified matrices. *Memorandum Report*, 42:336, 1950.

Y. Yao, P. Wang, B. Tian, S. Cheng, Z. Li, S. Deng, H. Chen, and N. Zhang. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*, 2023.