Sleep-Wake Consolidation for Lifelong Conversational Memory in Local Language Models

Vladimir Baranov
vlad@chatsetter.ai

Abstract

Large Language Models lack persistent memory: each session begins from a blank state, and all conversational context is lost when the session ends. Existing approaches to this problem—retrieval-augmented generation, summary injection, and external memory modules—keep the model's weights frozen, relying on input manipulation rather than genuine learning. We present a system that enables a local LLM to form long-term memories by integrating conversational experience directly into its weights through a biologically-inspired sleep-wake cycle. Drawing on Complementary Learning Systems theory, the system alternates between a wake phase (standard inference with conversation logging) and a sleep phase (a six-stage pipeline of curation, experience replay, synthetic data generation, LoRA fine-tuning, validation gating, and adapter fusion). We implement and evaluate the system on a 3-billion-parameter quantized model running on a MacBook Air with 8 GB of RAM using the MLX framework. Our experiments reveal a narrow but viable learning rate window (approximately 1×10^{-4}) for stable continual learning at this scale, outside of which the model either fails to learn or suffers catastrophic forgetting. Within this window, the model successfully transfers factual information from conversations into its weights, surviving complete restarts with no context window assistance. Successive sleep cycles strengthen recall through spaced repetition, consistent with predictions from the memory consolidation literature.

1 Introduction

Every modern large language model suffers from a fundamental limitation: it cannot learn from its own conversations. A user may spend hours sharing personal details, establishing preferences, and building context, but the moment the session ends, all of it vanishes. The next conversation starts from a blank slate. The context window provides an illusion of memory during a session—the model can reference earlier exchanges because those tokens remain in its attention mechanism—but this is working memory, not long-term memory. It has a hard size limit, disappears between sessions, and provides no mechanism for the model to actually learn from experience.

Several approaches have been proposed to address this gap. Retrieval-augmented generation (RAG) stores conversation snippets in an external database and injects relevant ones into the prompt at inference time (Lewis et al., 2020). Summary-based systems compress conversation history into condensed representations that persist across sessions. Memory-augmented architectures add external read-write memory modules to the model (Wang et al., 2024b, 2023a). Each of these approaches keeps the model's weights frozen—the model itself never changes, it simply receives different inputs. Recent comparisons suggest that RAG consistently outperforms unsupervised fine-tuning for factual knowledge injection (Ovadia et al., 2024), though the two approaches may be complementary rather than competing (de Luis Balaguer et al., 2024).

This paper takes a different approach. We ask: what if the model itself could learn from its conversations? Inspired by the Complementary Learning Systems (CLS) framework from neuroscience (McClelland et al., 1995; Kumaran et al., 2016), we design a system in which the context window serves as a fast, episodic store (analogous to the hippocampus) and the model's weights serve as a slow, semantic store (analogous to the neocortex). Periodically, the model "sleeps"—an offline consolidation cycle that absorbs recent conversations into the weights using Low-Rank Adaptation (LoRA) (Hu et al., 2022), with safeguards against catastrophic forgetting inspired by the brain's own mechanisms for memory consolidation during sleep.

We implement this system end-to-end on consumer hardware—a MacBook Air M3 with 8 GB of unified memory—and demonstrate that a 3-billion-parameter quantized language model can, after sleeping, recall facts from a prior conversation with no context window assistance. Our contributions are:

1. **A complete sleep-wake system for conversational memory.** We present the first end-to-end architecture integrating curation, experience replay with spaced repetition, synthetic data generation, LoRA training, and validation gating into a unified sleep-wake loop for persistent conversational memory in a local LLM.

2. **Empirical characterization of the viable learning rate window.** We identify a narrow band of hyperparameters (learning rate $\approx 1 \times 10^{-4}$, single epoch) that enables stable continual learning at 3B scale on consumer hardware, and document the failure modes on either side of this window.

3. **Evidence of spaced repetition effects across sleep cycles.** We demonstrate that successive consolidation cycles strengthen memory recall, consistent with predictions from the spaced repetition literature, providing evidence that the sleep-wake architecture produces emergent consolidation dynamics.

2 Related Work

2.1 Continual Learning for Language Models

Continual learning—the ability to acquire new knowledge without forgetting old knowledge—is a long-standing challenge in neural networks. Comprehensive surveys catalog the landscape for LLMs specifically (Wu et al., 2024; Shi et al., 2024), identifying three settings: continual pre-training, domain-adaptive pre-training, and continual fine-tuning. Our work falls in the continual fine-tuning category, where a model is updated on task-specific data after initial training.

Classical approaches to catastrophic forgetting include Elastic Weight Consolidation (EWC), which uses the Fisher information matrix to protect important parameters (Kirkpatrick et al., 2017); Learning without Forgetting (LwF), which uses knowledge distillation to preserve old-task performance (Li and Hoiem, 2017); and progressive neural networks, which add new capacity for each task (Rusu et al., 2016). Recent work on LLMs specifically has shown that forgetting intensifies with model scale during continual instruction tuning (Luo et al., 2023), that the flatness of the loss landscape directly influences forgetting severity (Li et al., 2024), and that apparent performance drops may sometimes reflect disrupted task alignment rather than true knowledge loss (Zheng et al., 2025). The TRACE benchmark reveals severe degradation when LLMs are trained sequentially on diverse tasks (Wang et al., 2024a).

Our system addresses catastrophic forgetting through a combination of low learning rates, LoRA-constrained updates, experience replay, and a validation gate that rolls back destructive sleep cycles—an integrated approach rather than a single mechanism.

2.2 Sleep-Inspired and Biologically-Motivated Approaches

The theoretical foundation for our work comes from Complementary Learning Systems (CLS) theory, which argues that the brain requires two learning systems: a hippocampal system for rapid episodic encoding and a neocortical system for gradual extraction of statistical structure (McClelland et al., 1995). The theory was updated to account for the role of hippocampal replay in generalization and the capacity for rapid neocortical learning when new information is consistent with existing schemas (Kumaran et al., 2016). The original wake-sleep algorithm (Hinton et al., 1995) used alternating phases for training generative models, though its connection to biological sleep consolidation was metaphorical rather than mechanistic.

Several recent works have operationalized sleep-inspired consolidation for neural networks. Tadros et al. (2022) interleave backpropagation with simulated sleep using Hebbian plasticity rules, showing that offline replay protects old memories during new task learning. Krishnan et al. (2019) convert trained ANNs to spiking networks for a sleep-like phase using spike-timing dependent plasticity. In the continual learning setting, Carta et al. (2024) introduce Wake-Sleep Consolidated Learning with explicit wake, NREM, and REM phases, outperforming baselines on image classification. Harun et al. (2023) propose SIESTA, a wake/sleep framework for on-device continual learning that matches offline learner performance on ImageNet-1K.

More recently, concurrent work has explored sleep for language models specifically: "Language Models Need Sleep" (Anonymous, 2025) proposes RL-based knowledge seeding and synthetic curriculum generation, while "Dreaming is All You Need" (Anonymous, 2024) incorporates sleep cycles into training through unsupervised learning features.

Our work differs from this body of literature in three respects. First, we target *conversational memory*—the ability to remember facts from natural dialogue—rather than task-incremental classification or benchmark performance. Second, we operate on consumer hardware under severe resource constraints (8 GB RAM, 3B parameters), which introduces unique challenges around the viable learning rate window. Third, we implement a complete end-to-end system rather than an isolated training algorithm, including curation, replay scheduling, validation gating, and checkpoint management.

2.3 Parameter-Efficient Fine-Tuning

Low-Rank Adaptation (LoRA) (Hu et al., 2022) freezes pre-trained weights and injects trainable low-rank decomposition matrices into transformer layers, reducing trainable parameters by orders of magnitude while matching full fine-tuning quality. QLoRA extends this by backpropagating through 4-bit quantized weights (Dettmers et al., 2023). Earlier parameter-efficient approaches include adapter modules (Houlsby et al., 2019) and prefix-tuning (Li and Liang, 2021).

LoRA has been specifically studied for continual learning. O-LoRA learns tasks in orthogonal low-rank subspaces to minimize inter-task interference (Wang et al., 2023b). InfLoRA designs interference-free subspaces that eliminate the effect of new tasks on old task representations (Liang and Li, 2024). Our system uses standard LoRA with adapter fusion after each sleep cycle rather than maintaining separate adapters per task, as the "tasks" in our setting (individual conversations) are not discrete or well-separated.

2.4 Experience Replay

Experience replay—storing and replaying past examples during training—is one of the oldest and most effective strategies for mitigating catastrophic forgetting. Gradient Episodic Memory (GEM) constrains gradient updates using stored examples (Lopez-Paz and Ranzato, 2017), with A-GEM providing a more efficient approximation (Chaudhry et al., 2019). Dark Experience Replay (DER++) replays stored logits alongside labels for stronger consistency (Buzzega et al., 2020). In the LLM setting, Rolnick et al. (2019) demonstrate that simple experience replay substantially reduces forgetting in reinforcement learning. Huang et al. (2024) propose Self-Synthesized Rehearsal (SSR), which uses the LLM itself to generate rehearsal examples from its own knowledge before fine-tuning, eliminating the need for stored training data.

Our replay buffer implements prioritized spaced repetition: high-scoring examples from previous sleep cycles are mixed into each training batch with a decay factor (0.85) that reduces their priority over successive cycles. This design is motivated by the spacing effect in memory research—repeated exposures with intervening periods of partial decay produce stronger encoding than massed repetition.

2.5 Memory-Augmented and Retrieval-Augmented LLMs

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) combines parametric models with non-parametric retrieval over external corpora, demonstrating improved factual accuracy on knowledge-intensive tasks. MemoryLLM introduces a self-updatable memory pool in the transformer's latent space that retains information across nearly a million updates (Wang et al., 2024b). LongMem uses a decoupled architecture with a frozen backbone and an adaptive side-network for retrievable long-term memory (Wang et al., 2023a). Systematic comparisons show that RAG outperforms naive fine-tuning for factual knowledge injection, though fine-tuning excels when the goal is domain adaptation rather than factual recall (Ovadia et al., 2024; de Luis Balaguer et al., 2024).

These approaches keep the model's weights frozen, treating memory as an external resource accessed through the input. Our approach is complementary: we modify the weights themselves, making the model's knowledge genuinely persistent and independent of any retrieval infrastructure. The finding by Ovadia et al. (2024) that exposure to multiple variations of the same fact improves fine-tuning effectiveness directly motivates our synthetic data generation ("dreaming") stage.

2.6 Self-Training and Synthetic Data Generation

Training on self-generated data has proven effective across several settings. Self-Instruct bootstraps instruction-following capabilities from a model's own generations (Wang et al., 2023c). SPIN uses self-play against previous model iterations to improve alignment (Chen et al., 2024). Constitutional AI uses the model's own critiques for self-improvement (Bai et al., 2022). Rho-1 demonstrates that selectively training on high-value tokens produces dramatically better outcomes than uniform training (Lin et al., 2024). Cheng et al. (2024) show that transforming raw corpora into reading comprehension format preserves prompting ability during domain adaptation.

Our system's "dreaming" stage generates synthetic Q&A pairs that approach learned information from multiple angles, building associative richness before training. This is related to SSR (Huang et al., 2024) and Self-Instruct (Wang et al., 2023c), but applied specifically to consolidate conversational memories rather than to generate general training data.

3 Method

The system is organized as a state machine alternating between two phases: waking (inference) and sleeping (training). An orchestrator manages transitions, triggered either automatically after a configurable number of conversational turns or manually by the user.

3.1 System Overview

The architecture implements a dual-system design following CLS theory. The context window acts as the fast-learning system (hippocampal analog), rapidly encoding new conversational exchanges. The model weights act as the slow-learning system (neocortical analog), gradually integrating experience during offline consolidation. The sleep-wake loop mediates the transfer between these two systems.

The system comprises four modules: the **wake module** (chat loop, context management, conversation logging), the **sleep module** (curation, training, validation, dreaming), the **memory module** (replay buffer, checkpoints, identity reinforcement), and an **orchestrator** that manages state transitions.

3.2 Wake Phase

During the wake phase, the system operates as a standard chat interface with three concurrent subsystems:

Context management. A sliding window manages the tokens available to the model's attention mechanism. When the window reaches 80% capacity, older messages are summarized by the model itself and replaced with a compressed representation. This mirrors working memory refresh—older information is abstracted while recent details remain available.

Conversation logging. Every exchange is persisted to disk in JSONL format, providing the raw material for sleep-phase processing. Unlike biological memory, the log provides a perfect record with no degradation or distortion.

Sleep trigger monitoring. A turn counter tracks conversational depth. When it reaches a configurable threshold (default: 10 turns), the system transitions to the sleep phase. Manual triggering is also supported.

3.3 Sleep Phase

Sleep is a six-stage pipeline that transforms raw conversation into weight updates.

Stage 1: Curation. The conversation log is scored along three dimensions. For each exchange e, we compute:

- novelty(e): the degree to which the information is not already represented in the model's knowledge;
- importance(e): the relevance and significance of the information (explicit user corrections, stated preferences, and novel factual content score higher);
- utility(e): the anticipated future usefulness of the information.

Exchanges below configurable thresholds on these dimensions are discarded. This filtering mirrors the brain's selective consolidation during the transition from waking to sleep, where the hippocampus preferentially consolidates memories anticipated to be useful.

Stage 2: Replay buffer integration. High-scoring examples from previous sleep cycles are mixed into the training data at a configurable ratio (default: 20%). Each time an item is replayed, its priority is reduced by a decay factor $d = 0.85$:

$$\text{priority}_t(e) = \text{priority}_{t-1}(e) \cdot d \tag{1}$$

This implements spaced repetition: important information is reinforced across multiple sleep cycles with declining frequency, following the spacing effect observed in human memory research.

Stage 3: Dreaming. During deep sleep cycles (every k light sleep cycles, default $k = 5$), the system enters a REM-equivalent phase. The model generates synthetic Q&A pairs based on its accumulated knowledge, creating new associative connections. For example, if the model has learned that the user works with PostgreSQL and previously discussed connection pooling, the dreamer might generate training pairs about PostgreSQL connection pooling best practices—strengthening the association between related memories. This is analogous to the creative recombination function attributed to REM sleep in neuroscience.

Stage 4: LoRA training. The curated dataset is used to train a Low-Rank Adaptation layer. Following Hu et al. (2022), we inject trainable low-rank matrices $\mathbf{A} \in \mathbb{R}^{d \times r}$ and $\mathbf{B} \in \mathbb{R}^{r \times d}$ into the model's attention layers, where $r \ll d$ is the rank. The weight update is constrained to the low-rank subspace:

$$\mathbf{W}' = \mathbf{W} + \mathbf{BA} \tag{2}$$

This constrains the update to a low-dimensional subspace, minimizing interference between new and existing knowledge. Training runs for N iterations scaled to dataset size:

$$N = |\mathcal{D}| \times \text{epochs} \tag{3}$$

where $|\mathcal{D}|$ is the number of training examples and epochs is typically 1 (a single pass).

Stage 5: Validation. Before and after training, the model is evaluated on a fixed set of benchmark questions $\mathcal{Q} = \{q_1, \ldots, q_n\}$. Let s_{pre} and s_{post} denote the pre-sleep and post-sleep scores respectively. The sleep cycle is accepted only if:

$$s_{\text{post}} \geq \tau \cdot s_{\text{pre}} \tag{4}$$

where τ is the validation threshold (default: 0.5). If validation fails, the LoRA adapter is discarded and the model reverts to its pre-sleep state. Critically, fusion occurs only *after* validation passes—an ordering learned through a failure where fusing before validation left the system unable to recover from a destructive training cycle.

Stage 6: Fusion. If validation passes, the LoRA adapter is merged into the base model weights and saved as a new checkpoint. The system reloads the updated model and resumes the wake phase. The conversation has become part of the model's knowledge.

3.4 Multi-Timescale Architecture

The system operates at multiple timescales, mirroring the multi-stage consolidation hierarchy observed in biological sleep:

Table 1: Multi-timescale sleep architecture. Each layer trades off plasticity against stability, with deeper layers performing more thorough but less frequent consolidation.

Layer	Human Analog	LLM Implementation	Frequency
Layer 1	Working memory	Context window, no weight changes	Every turn
Layer 2	Ultradian dips	Small adapter updates or memory store writes	∼15 turns
Layer 3	NREM Stages 1–2	LoRA fine-tune on curated session data	End of session
Layer 4	Slow-wave sleep	Full consolidation with replay, adapter fusion	Daily
Layer 5	REM	Synthetic Q&A generation, creative association	During deep sleep

The key design principle is that plasticity and stability operate on a spectrum: frequent, light updates provide rapid adaptation with low risk, while infrequent, deep updates provide thorough integration at higher risk. Operating at multiple timescales simultaneously balances these pressures.

3.5 Identity Reinforcement

The system maintains identity through two mechanisms at different timescales. The **system prompt** is a plain-text string injected at the start of every inference call—it exists only in the context window and takes effect immediately. The **identity dataset** is a collection of core Q&A pairs (e.g., "What is your name?" → "My name is J") that are included in every sleep cycle's training data. This serves as a form of core memory reinforcement, analogous to the deeply rehearsed self-knowledge that forms the most stable layer of human memory, preventing identity drift across successive sleep cycles.

4 Experimental Setup

4.1 Hardware and Software

All experiments were conducted on a MacBook Air M3 with 8 GB of unified memory. Apple Silicon's unified memory architecture, where CPU, GPU, and Neural Engine share the same RAM, makes it suited for local LLM workloads but imposes hard constraints: after accounting for the operating system (∼3 GB), approximately 5 GB remains for the model, inference, and training. The system uses Apple's MLX framework (Hannun et al., 2023) for both inference and LoRA training, leveraging unified memory to avoid CPU–GPU transfer bottlenecks.

4.2 Model and Hyperparameters

We use Llama 3.2 3B Instruct at 4-bit quantization,[1] which requires approximately 1.8 GB on disk and 2.5 GB in RAM, leaving sufficient headroom for LoRA training. Table 2 summarizes the hyperparameter configuration.

[1] `mlx-community/Llama-3.2-3B-Instruct-4bit`

Table 2: Hyperparameter configuration for the sleep-wake system.

Parameter	Value	Rationale
Learning rate	1×10^{-4}	Midpoint of viable window (Section 5.1)
Epochs	1 (single pass)	Each example seen exactly once per cycle
LoRA rank (r)	16	Moderate adapter capacity
LoRA alpha	32	Effective scaling of 2.0 (α/r)
LoRA layers	8	Eight transformer layers modified
Batch size	1	Memory constraint on 8 GB hardware
Validation threshold (τ)	0.5	Post-sleep score must exceed 50% of pre-sleep
Validation questions	5	Reduced from 20 for speed
Replay ratio	0.2	20% of training batch from replay buffer
Replay decay (d)	0.85	Priority reduction per replay

4.3 Evaluation Protocol

We evaluate along two axes. **General capability preservation** is measured using a fixed set of 5 benchmark questions administered before and after each sleep cycle, scoring the model's ability to produce coherent and accurate responses. **Memory formation** is tested by introducing novel factual information during conversation that the model could not know from pretraining, then restarting the application (clearing the context window entirely) and querying the model about that information with no context clues.

5 Results

5.1 Learning Rate Sensitivity

Systematic exploration of the hyperparameter space reveals a narrow band of viability for the 3B model on constrained hardware (Table 3).

Table 3: Learning rate sensitivity. The viable window spans approximately one order of magnitude. Configurations above 1×10^{-4} cause catastrophic forgetting; configurations below produce no measurable learning.

Learning Rate	Epochs	Iterations	Result
5×10^{-4}	5	∼500	Total destruction (benchmark: 0.00)
2×10^{-4}	3	∼276	Catastrophic forgetting (benchmark: 0.00)
5×10^{-5}	3	∼270	No measurable learning, no damage
$\mathbf{1 \times 10^{-4}}$	**1**	**∼90**	**Success: learned, retained general ability**

The viable window is remarkably narrow. One order of magnitude above the working learning rate destroys the model entirely; one order below produces no measurable effect. At the destructive end (5×10^{-4} with 5 epochs), each training example was seen approximately 15 times at a learning rate appropriate for training from scratch, overwriting pretrained knowledge entirely. At the inert end (5×10^{-5}), the gradient updates were too small to register on a 3-billion-parameter model. The successful configuration—a single pass at 1×10^{-4}—provides just enough signal to encode new information without destabilizing existing representations.

This finding has implications for scaling: larger models should have a wider viable window, as the same gradient update distributes across more parameters and causes proportionally less

disruption per weight.

5.2 Memory Formation

The memory formation test introduced a completely fabricated fact that the model could not know from pretraining: detailed biographical information about a fictional person, including personal relationships and career details. This information was conveyed through natural conversation.

After one sleep cycle with the successful hyperparameter configuration, the application was restarted, clearing the context window entirely. When queried about the fictional person with no context clues, the model correctly identified the core biographical fact—the detail that appeared most frequently across training examples (raw conversation pairs, extracted Q&A pairs, and replay buffer entries). The model did not recall the specific relationship detail, which appeared in fewer training variations.

This result is consistent with repetition-dependent memory consolidation: facts encountered across more training examples survived the consolidation process, while less-repeated details did not.

5.3 Spaced Repetition Effect

After a second sleep cycle, recall of the injected information improved. The replay buffer resurfaced the target information during the second training pass at reduced priority (decay factor 0.85), strengthening the encoding through spaced exposure. This progressive strengthening across cycles—rather than single-shot memorization—demonstrates that the architecture produces emergent consolidation dynamics consistent with the spaced repetition literature.

The full sleep cycle with the working configuration required approximately 3–5 minutes on the 8 GB MacBook Air.

6 Discussion

6.1 The Narrow Viability Window and Implications for Scale

The narrow learning rate window (approximately one order of magnitude) at 3B scale suggests that model capacity is a binding constraint. A 70-billion-parameter model has over twenty times the parameter count; the same LoRA update that barely registered—or caused catastrophic forgetting—on the 3B model would distribute across vastly more weights, producing proportionally gentler updates. We expect larger models to exhibit a substantially wider viable window, enabling more aggressive learning rates and deeper consolidation per cycle.

Table 4: Expected scaling behavior across hardware configurations.

Hardware	**RAM**	**Model**	**Expected Behavior**
MacBook Air M3	8 GB	3B 4-bit	Narrow viable window; partial recall
Mac Mini M4	16–32 GB	8B 4-bit	Wider window; $\sim$2.5$\times$ more parameters
Mac Studio M4 Ultra	128–192 GB	70B 4-bit	Full knowledge absorption; robust recall

6.2 Where the Biological Analogy Holds and Breaks

The CLS framework proved productive as a design tool: dual learning rates, spaced repetition, curation, and dreaming all map directly to neuroscience concepts and led to working engineering decisions. The analogy breaks in instructive ways. Biological learning is continuous—synaptic changes happen in real time during waking experience, with sleep serving as reorganization rather than the sole site of learning. Our system's sharp boundary between inference (no weight changes) and training (no inference) is an engineering constraint imposed by current frameworks, not a theoretical preference. The brain also employs massive parallelism and architectural diversity, with distinct systems for episodic, semantic, procedural, and emotional memory. Our system stores everything in a single LoRA adapter with uniform training dynamics.

6.3 Limitations

Training on model outputs. The system trains on the full conversation, including the model's own responses. Hallucinated content gets reinforced. A future version should weight user-provided ground truth differently from model-generated responses.

No deduplication across cycles. The system currently gathers all conversations at every sleep cycle, including those already trained on. Early conversations receive disproportionate reinforcement, risking overfitting to initial interactions.

Shallow curation. The keyword-based scoring heuristics lack genuine understanding of importance. Model-based curation—using the LLM itself to evaluate significance—would be more effective but computationally expensive on constrained hardware.

No selective forgetting. Once information is integrated into the weights, there is no mechanism to remove it short of rolling back to a prior checkpoint. Biological systems have active forgetting mechanisms; our system does not.

Blocking sleep. The model goes offline during sleep. A production system would require background training on a model copy or a secondary model for handling requests during consolidation.

Limited evaluation scale. Our results demonstrate the mechanism on a single test case. Comprehensive evaluation with larger fact sets, multiple domains, and longer time horizons is needed to characterize the system's capacity and failure modes.

7 Future Work

The most immediate priority is scaling to larger models to test whether the viable learning rate window widens as predicted. A secondary priority is replacing keyword-based curation with model-based importance scoring that can distinguish genuinely novel information from routine exchanges.

The dreaming mechanism warrants deeper exploration. A more sophisticated approach would have the model actively search for contradictions in its own knowledge, generate scenarios that test the boundaries of what it has learned, and use these self-generated challenges as training signal—closer to the creative recombination function attributed to REM sleep.

Additional directions include multi-user support with separate memory partitions, selective forgetting mechanisms, non-blocking background training, separation of factual and behavioral

learning into distinct adaptation pathways, and online learning that updates weights during inference, eliminating the sharp wake-sleep boundary and moving toward the brain's continuous learning regime.

8 Conclusion

We presented a system that enables a local language model to form persistent conversational memories through biologically-inspired sleep-wake cycles. A 3-billion-parameter model on consumer hardware successfully transferred factual information from a conversation into its weights, surviving complete restarts with no context window assistance. Successive sleep cycles strengthened this memory through spaced repetition. The engineering path to this result revealed a narrow viable hyperparameter window—a finding with practical implications for anyone attempting continual learning on resource-constrained hardware. The gap between the theoretical framework and a working system contained multiple distinct failure modes, each requiring targeted fixes. The viable parameter space was narrow, a reminder that balancing plasticity and stability in artificial systems remains a nontrivial optimization problem even when the theory is sound.

References

Anonymous. Dreaming is all you need. *arXiv preprint arXiv:2409.01633*, 2024.

Anonymous. Language models need sleep: Learning to self modify and consolidate memories. *OpenReview*, 2025.

Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*, volume 33, pages 15920–15930, 2020.

A. Carta et al. Wake-sleep consolidated learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. Also arXiv:2401.08623.

A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations*, 2019.

Z. Chen, Y. Deng, H. Yuan, K. Ji, and Q. Gu. Self-play fine-tuning converts weak language models to strong language models. In *International Conference on Machine Learning*, 2024.

D. Cheng, S. Huang, and F. Wei. Adapting large language models to domains via reading comprehension. In *International Conference on Learning Representations*, 2024.

M. A. de Luis Balaguer, V. Benara, R. L. de Freitas Cunha, et al. RAG vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture. *arXiv preprint arXiv:2401.08406*, 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.

A. Hannun, J. Digani, A. Katharopoulos, and R. Collobert. MLX: Efficient and flexible machine learning on Apple silicon. Apple Machine Learning Research. https://github.com/ml-explore/mlx, 2023.

M. Y. Harun, J. Gallardo, T. L. Hayes, R. Kemker, and C. Kanan. SIESTA: Efficient online continual learning with sleep. *Transactions on Machine Learning Research*, 2023.

G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799, 2019.

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. Huang, L. Cui, A. Wang, C. Yang, X. Liao, L. Song, J. Yao, and J. Su. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

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. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

G. P. Krishnan, T. Tadros, R. Ramyaa, and M. Bazhenov. Biologically inspired sleep algorithm for artificial neural networks. *arXiv preprint arXiv:1908.02240*, 2019.

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.

X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 4582–4597, 2021.

Z. Li and D. Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017.

Y.-S. Liang and W.-J. Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23638–23647, 2024.

Z. Lin, Z. Gou, Y. Gong, X. Liu, Y. Shen, R. Xu, C. Lin, Y. Yang, J. Jiao, N. Duan, and W. Chen. Rho-1: Not all tokens are what you need. In *Advances in Neural Information Processing Systems*, 2024. Oral, Best Paper Runner-Up.

D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

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.

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.

O. Ovadia, M. Brief, M. Mishaeli, and O. Elisha. Fine-tuning or retrieval? comparing knowledge injection in LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

H. Shi, Z. Xu, H. Wang, W. Qin, W. Wang, Y. Wang, and H. Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 2024.

T. Tadros, G. P. Krishnan, R. Ramyaa, and M. Bazhenov. Sleep-like unsupervised replay reduces catastrophic forgetting in artificial neural networks. *Nature Communications*, 13:7742, 2022.

W. Wang, L. Dong, H. Cheng, X. Liu, X. Yan, J. Gao, and F. Wei. Augmenting language models with long-term memory. In *Advances in Neural Information Processing Systems*, volume 36, 2023a.

X. Wang, T. Chen, Q. Ge, H. Xia, R. Bao, R. Zheng, Q. Zhang, T. Gui, and X. Huang. O-LoRA: Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671, 2023b.

X. Wang, Y. Zhang, T. Chen, S. Gao, S. Jin, X. Yang, Z. Xi, R. Zheng, Y. Zou, T. Gui, Q. Zhang, and X. Huang. TRACE: A comprehensive benchmark for continual learning in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024a. arXiv:2310.06762.

Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023c.

Y. Wang, Y. Gao, X. Chen, H. Jiang, S. Li, J. Yang, Q. Yin, Z. Li, X. Li, B. Yin, J. Shang, and J. McAuley. MemoryLLM: Towards self-updatable large language models. In *International Conference on Machine Learning*, 2024b.

T. Wu, L. Luo, Y.-F. Li, S. Pan, T.-T. Vu, and G. Haffari. Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364*, 2024.

J. Zheng, X. Cai, S. Qiu, and Q. Ma. Spurious forgetting in continual learning of language models. In *International Conference on Learning Representations*, 2025.