LANGUAGE MODELS NEED SLEEP: LEARNING TO SELF-MODIFY AND CONSOLIDATE MEMORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

The past few decades have witnessed significant advances in the design of machine learning algorithms– from early studies of task-specific shallow models to more general deep Large Language Models (LLMs). Despite showing promising results on tasks requiring instant prediction or in-context learning, existing models lack the ability to continually learn and effectively transfer their temporal in-context knowledge into their long-term parameters. Inspired by the human learning process, we introduce a "*Sleep*" paradigm that enables models to continually learn, distill their fragile short-term memories into stable long-term knowledge through replay, and recursively improve themselves through a "*Dreaming*" process. More specifically, Sleep consists of two stages: (1) **Memory Consolidation**: an *upward* distillation process, called *Knowledge Seeding*, in which the memories of a *smaller* model are distilled into a *larger* network to provide greater capacity, while preserving the knowledge. As a proof of concept, we present a new *Generalized Distillation* process for Knowledge Seeding, combining on-policy distillation with Reinforcement Learning (RL)-based imitation learning); and (2) **Dreaming**: a self-improvement phase, in which the model uses RL to generate a curriculum of synthetic data for rehearsing new knowledge and refining existing capabilities without human supervision. We provide rigorous theoretical analysis showing that the Knowledge Seeding objective admits a closed-form truncated SVD solution (under a linear upper network) in a space jointly shaped by the student's on-policy covariance and a reward metric tensor. We also show that random MoE expert injection during dreaming provably improves the conditioning of the gradient covariance. Our experiments on long-horizon, continual learning, knowledge incorporation, and few-shot generalization tasks support the importance of the Sleep stage.

1 INTRODUCTION

The development of Large Language Models (LLMs) marks a pivotal milestone in machine learning research: a paradigm shift from task-specific models to more general-purpose systems with various emergent capabilities (Brown et al., 2020; Schaeffer et al., 2023). Despite LLMs' remarkable capabilities in diverse sets of tasks (Wang et al., 2023; Nijkamp et al., 2023; Comanici et al., 2025), they are largely static after their initial deployment, unable to *continually acquire* new capabilities beyond their immediate context, operating with a fixed "knowledge cutoff" date (Cheng et al., 2024). Efforts to overcome this limitation have primarily focused on: (1) re-pretraining on an expanded dataset, which is computationally expensive and impractical for frequent updates (Ibrahim et al., 2024); (2) continual parameter updates such as fine-tuning or low-rank adaption (Hu et al., 2022; Akyürek et al., 2024a), which often triggers Catastrophic Forgetting (CF) (Kemker et al., 2018; Shi et al., 2024)—a well-known phenomenon where proficiency on original tasks degrades as the model learns new ones.

In-Context Learning (ICL) (Brown et al., 2020) has emerged as an efficient form of continual learning (Akyürek et al., 2022; Dong et al., 2024; Akyürek et al., 2024b; Li et al., 2025), where the model performs internal meta-learning computations to incorporate context knowledge into a short-term memory (Behrouz et al., 2025; Dherin et al., 2025). However, ICL is limited to the context window: any acquired knowledge is discarded at session's end. As noted by Behrouz et al. (2025),

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

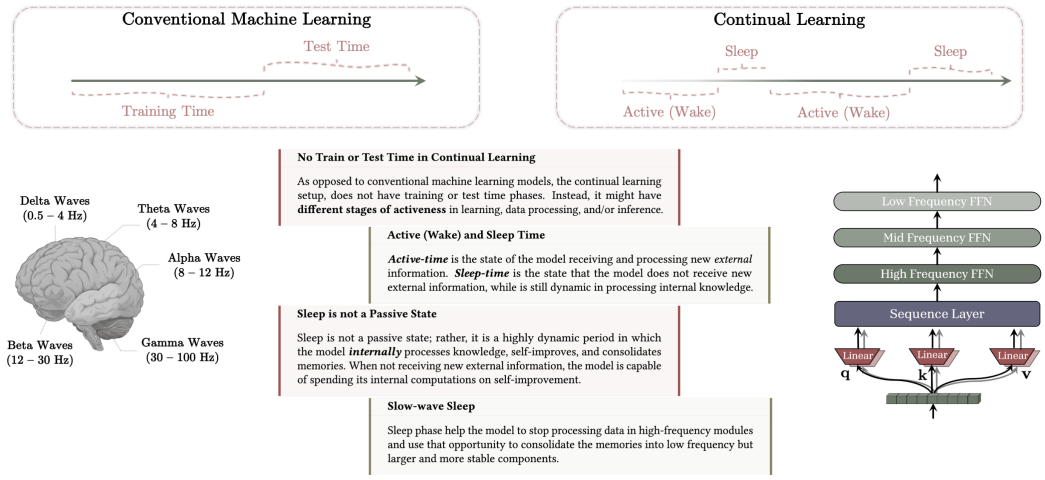


Figure 1: (**Conventional Machine Learning vs. Continual Learning**) While in conventional machine learning often the lifespan of the model is divided to *test and training time*, continual learning setup does not have these phases. We suggest that a continual learner need to have different stages of activeness in learning, which we refer to as: (i) *Active or Wake Time*, and (ii) *Sleep Time*. Sleep time is not a passive state, rather it internally process the data to consolidate the memories from fast unstable modules to more stable low frequency (slow) components.

this resembles anterograde amnesia (Scoville & Milner, 1957)—an LLM's knowledge is limited to either: (1) the immediate context (ICL), or (2) MLP layers storing long-past pre-training knowledge. This raises a critical question: *How can the model transfer fragile short-term memories into stable long-term knowledge*?

The human brain addresses this through neuroplasticity (Pascual-Leone et al., 2005; Johnston, 2009) and two complementary consolidation processes (Goto et al., 2021; Frey & Morris, 1997; Yang et al., 2024): (1) rapid "online" consolidation during wakefulness, and (2) "offline" systems consolidation during sleep via replay of recently encoded patterns (Ji & Wilson, 2007; Peyrache et al., 2009; Foster & Wilson, 2006). Sleep is not a passive state but a dynamic period of brain activity essential for cognition (Rasch & Born, 2013; Goldstein & Walker, 2014; Wamsley & Stickgold, 2011). Non-REM (NREM) sleep performs memory consolidation—transforming fragile experiences into stable knowledge through hippocampal-neocortical dialogue (Squire & Alvarez, 1995; Squire et al., 2015)—and synaptic homeostasis (Tononi & Cirelli, 2006). REM sleep, associated with dreaming, selectively strengthens newly formed synapses and integrates new information with existing knowledge. The cyclical NREM→REM alternation first consolidates and prunes experiences, then explores novel connections and strengthens salient pathways.

1.1 Contributions

Inspired by the memory processing in humans, we argue that for a continual learner, there is no train or test time. Instead, the model needs to periodically have two phases of being "active" or "Sleep"; In the active state, the model receives new data and processes it, while in the sleep phase the focus is on the internal knowledge and the consolidation of recent memories. To this end, we introduce an instance of "sleep" paradigm for LLMs that consists of two integrated phases:

i **Memory Consolidation**: The model periodically activates new parameters and distills knowledge from higher-frequency (faster updating) modules to newly unlocked parameters in more stable (lower-frequency) modules, allowing plasticity while preserving old knowledge.

ii **Self-Improvement via Dreaming**: The model generates synthetic data (dreams) to recursively improve its own performance, with particular focus on recently added knowledge.

From the technical point of view, our contributions to each of the above phases are:

1. **Periodic Parameter (De)Activation**: Building on the Nested Learning (NL) paradigm (Behrouz et al., 2025), we suggest a periodic and gradual parameter (de)activation process that maintains plasticity while avoiding knowledge interference with previous parameters.
2. **Knowledge Seeding (Upward Distillation)**: We present a new form of knowledge transfer, called knowledge seeding, where one or some ***smaller*** models distill their knowledge to a ***larger*** model. We then use self-Knowledge Seeding (SKS) as a solution for memory consolidation.
3. **Generalized Knowledge Distillation (GKD) with Imitation Learning**: The formulation of SKS is general, and any form of objective and knowledge transfer method can be used. Here, we present a new objective that combines on-policy distillation with an imitation learning process.
4. **Theoretical Analysis**: We show that the Knowledge Seeding objective admits a closed-form truncated SVD solution (Theorem 1), that random MoE expert injection improves gradient conditioning (Theorem 2), and that the combined injection–reward pipeline strictly reduces the condition number (Theorem 3). See Section 4 and Appendix A for details.
5. **Experimental Evaluation**: We evaluate the effectiveness of sleep paradigm on a set of challenging downstream tasks: (1) Factual Knowledge Incorporation; (2) Few-shot Learning; (3) Long-context Understanding; and (4) Continual Learning. The results support the importance of growing parameters with iterative knowledge distillation for continual learning.

2 Preliminaries and Problem Formulation

2.1 Notation

We use bold lowercase (resp. uppercase) letters for vectors (resp. matrices) and use subscript t to refer to the state of the entities correspond to time t. Superscripts for parameters of a module (resp. hyperparameters) are used to determine the update frequency of the module (resp. distinguish different instances). Through the paper, we let $x \in \mathbb{R}^{L \times d_{\text{in}}}$ be the input, $\mathbf{K}$ be the keys, $\mathbf{V}$ be the values, $\mathbf{Q}$ be the query matrices in the sequence model, and L denote the sequence length. When it is needed, we parameterize the language model $\texttt{LM}_\theta$ with $\theta = \{W_1^{(f_1)}, \dots, W_{k_1}^{(f_1)}\} \cup \{W_1^{(f_2)}, \dots, W_{k_2}^{(f_2)}\} \cup \dots \{W_1^{(f_c)}, \dots, W_{k_c}^{(f_c)}\}$, where parameter sets are sorted based on their weight update frequencies $f_1 \geq \dots, \geq f_c$ (see Definition 1).

2.2 Continuum Memory System

Transformer architectures consist of two critical components: (1) Attention module that acts as an associative memory and conditions the output on the past tokens in the context, which also results in in-context learning ability; and (2) MLP or feedforward layers, which are fixed after the training phase and encodes the knowledge acquired over the pre-training. As discussed by Behrouz et al. (2025), one can interpret such architectures as two-level memory systems, in which the attention's update span is the context length–meaning that at the end of the context, its corresponding parameters are updated and the acquired knowledge is forgotten–and MLP's update span is non-existence–indicating no update after pre-training. From this perspective, these two components are two extreme sides of the frequency spectrum, where the attention (resp. MLP) has infinite (resp. zero) update frequency.

Building on this intuition, Behrouz et al. (2025) presented Continuum Memory System (CMS), where the architecture is a sequence model such as attention, followed by a chain of MLP layers, each of which updated with its own frequency. More specifically, the time for one step of update in the slowest module is considered as the unit of time, and so the update rate of other components are defined as:

Definition 1 (Update Frequency)**.** *For any weight component of W, we define its frequency, denoted as f_W, as its number of updates per unit of time.*

To better understand this concept, we use a simple example of Fast-weight Programs (Schmidhuber, 1992), where the input is a sequence of length L. In this case, for each step of slow-weight (the unit of time), the fast-weight is updated L times, resulting in an update frequency of L.

Following this definition of frequency, CMS is formalized as a chain of MLP blocks $\texttt{MLP}^{(f_1)}(\cdot), \dots, \texttt{MLP}^{(f_k)}(\cdot)$, each of which is associated with a chunk size of $C^{(\ell)} := \frac{\max_{\ell'} C^{(\ell')}}{f_\ell}$

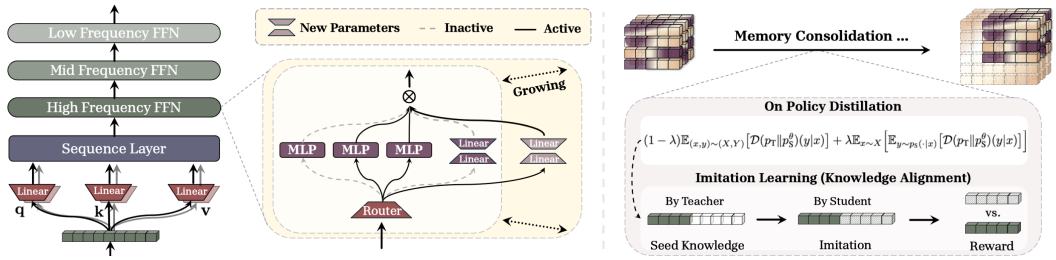


Figure 2: An overview of Memory Consolidation. The model increases its own number of parameters to enhance its capacity (Section 3.2), and then using the knowledge seeding, it transfers the knowledge abstractions from the higher to a lower-frequency memory (Section 3.3).

such that given input $x = \{x_1, \ldots, x_T\}$, with each x_t as one of the input data, the output of the chain is calculated as (we disregard normalizations for the sake of clarity):

$$y_t = \texttt{MLP}^{(f_k)}(\texttt{MLP}^{(f_{k-1})}(\cdots \texttt{MLP}^{(f_1)}(x_t))), \tag{1}$$

where the parameters of ℓ-th MLP block, denoted by $\boldsymbol{\theta}^{(f_\ell)}$, are updated every $C^{(\ell)}$ steps: $\boldsymbol{\theta}^{(f_\ell)}_{i+1} = \boldsymbol{\theta}^{(f_\ell)}_{i} - \boldsymbol{e}_{i,\ell}$, where:

$$\boldsymbol{e}_{i,\ell} = \begin{cases} \sum_{t=i-C^{(\ell)}}^{i} \eta_t^{(\ell)} f(\boldsymbol{\theta}_t^{(f_\ell)}; x_t) & \text{if } i \equiv 0 \pmod{C^{(\ell)}}, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Here $\eta_t^{(\ell)}$ are learning rates corresponds to $\boldsymbol{\theta}^{(f_\ell)}$, and $f(\cdot)$ is the error component of an arbitrary optimizer in an end-to-end optimization. Due to the generality of this formulation and being the superset of Transformers design (i.e., 1 MLP blocks will be equivalent to Transformers), throughout the paper, we use CMS as the default building blocks of the architectures. Also, for the sake of clarity and without loss of generality, we assume that $C^{(\ell)}$ is divisible by $C^{(\ell-1)}$.

From this perspective, sequence models (e.g., attention (Vaswani et al., 2017) or other memory modules or RNNs (Katharopoulos et al., 2020; Behrouz et al., 2024)) act as short-term memory, where high-frequency updates overwrite old knowledge to make room for new input. Conversely, CMS blocks form a memory spectrum: earlier blocks (higher frequency) store short-term memories, while later blocks store longer-term ones. Although active updates improve catastrophic forgetting (CF) resistance, CF can still occur when update periods synchronize (Behrouz et al., 2025). Thus, before updating any memory block, a mechanism must consolidate its abstracted knowledge into stable parameters. Reviewing sleep-inspired literature (Appendix F), we argue that unlike the conventional train/test paradigm, lifelong adaptation demands a continuous "wake-sleep" lifecycle.

3 The Sleep Paradigm

3.1 Learning Phases in Continual Learning

A continual learner is always learning from data/experience, so the conventional train/test split does not directly apply. We propose "active (wake)" and "sleep" time as the two phases of a continual learner's lifecycle. Following the discussion in Nested Learning (NL) (Behrouz et al., 2025), memory consolidation takes two forms: (1) *Online consolidation* occurs through end-to-end learning in the wake phase, where earlier (higher-frequency) layers transfer knowledge to later (lower-frequency) layers; (2) *Offline consolidation* occurs in the sleep phase, where the model stops processing new data and uses self-generated data to consolidate recent memories. We divide the sleep process into two key stages: (1) Memory consolidation; and (2) Dreaming for self-improvement. We summarize the full lifecycle in Algorithm 1 (Appendix C).

3.2 Memory Consolidation: Parameter Expansion

As discussed earlier, in memory consolidation, we aim to transfer short-term fragile memories into more vast, stable parameters (Algorithm 2, Appendix C). One of the important messages in CMS

formulation is: the fragility and/or stability of memories are relative. That is, for each memory block, other higher frequency memories are shorter-term and more fragile. Therefore, memory consolidation is not a simple two-step process, but an iterative operation that repeatedly transfers the knowledge stored in higher frequency memories into more stable lower-frequency parameters.

To avoid losing the knowledge of a faster updating block; an example of which is in-context learning (Brown et al., 2020), we need to perform memory consolidation step before updating its parameters. Therefore, given the list of chunk lengths $\{C^{(1)}, \dots, C^{(k)}\}$, the sleep process and so memory consolidation happens only at $\{C^{(1)} \times b, \dots, C^{(k)} \times b\}$ steps for all $b \in \mathbb{N}$. This process can be a critical bottleneck to unlock continual learning capabilities of LLMs, due to Catastrophic Forgetting (CF). This phenomenon is an inherent cause of model's limited capacity (e.g., number of parameters), where parameters need to be overridden to incorporate the new knowledge. Inspired by neuroplasticity, we present an efficient gradual parameter expansion in memory blocks that allows the model to shape new connections and so increases its own capacity.

Without loss of generality, we assume that the MLP blocks $\{\texttt{MLP}^{(f_\ell)}(\cdot)\}_{\ell=1}^{k}$ are sparse mixture of experts (MoEs) with a router $\mathcal{R}^{(f_\ell)}$: i.e., each $\texttt{MLP}^{(f_\ell)}(\cdot)$ includes a set of experts $\{W^{(f_\ell),1}, \cdots, W^{(f_\ell),\mathbf{s}_\ell}\}$, where $\mathbf{s}_\ell \geq 1$ is the current number of experts in the ℓ-th block of the chain. Let $(\ell^* - 1)$ be the index of the memory (or MLP) that we aim to consolidate its knowledge to its immediate next more stable memory module with index ℓ^*. To avoid the interference of transferred and previously stored knowledge in $\texttt{MLP}^{(f_{\ell^*})}(\cdot)$, we add a new low-rank expert to its set of parameters. That is, we add a low-rank MLP parametrized by $\{\mathbf{A}^{(f_{\ell^*}),\mathbf{s}_{\ell^*}+1}, \mathbf{B}^{(f_{\ell^*}),\mathbf{s}_{\ell^*}+1}\}$, where $\mathbf{A}^{(f_{\ell^*})} \in \mathbb{R}^{d \times d_{\text{low}}}$ and $\mathbf{B}^{(f_{\ell^*})} \in \mathbb{R}^{d_{\text{low}} \times d}$ ($d_{\text{low}} \ll d$), to the set of experts. Given this process, after each sleep time, the parameters of a subset of layers are growing.

3.3 Memory Consolidation with Knowledge Seeding

In this step, we aim to transfer the knowledge of $\texttt{MLP}^{(f_{\ell^*-1})}(\cdot)$ with parameters $\boldsymbol{\theta}^{(f_{\ell^*-1})}$ into the expanded set of parameters in $\texttt{MLP}^{(f_{\ell^*})}(\cdot)$ (see Algorithms 3 and 4 in Appendix C, for pseudo-code). Since sleep is triggered when the number of past steps is divisible by C^{ℓ^*-1}, the sender block $\texttt{MLP}^{(f_{\ell^*-1})}(\cdot)$ is scheduled for a base-weight update at this time. The consolidation procedure follows a compute–consolidate–update protocol:

(a) *Compute.* The base-weight update for the sender is computed from the accumulated gradients (equation 2), yielding prospective new parameters $\boldsymbol{\theta}^{(f_{\ell^*-1})}$. Crucially, this update is computed but not yet applied to the running model.

(b) *Consolidate.* We define:

- **Teacher ($\texttt{LM}_{\boldsymbol{\theta}}$):** the state of the model before the base-weight update or parameter expansion — i.e., $\texttt{MLP}^{(f_{\ell^*})}(\cdot)$ still uses its original parameters $\boldsymbol{\theta}^{(f_{\ell^*-1})}$ plus its accumulated experts;
- **Student ($\texttt{LM}_{\boldsymbol{\theta}_{\text{exp}}}$):** the state of the model using the prospective updated parameters $\boldsymbol{\theta}_{\text{new}}^{(f_{\ell^*-1})}$ (with experts in $\texttt{MLP}^{(f_{\ell^*-1})}(\cdot)$ reset), and with a newly added low-rank expert A, B in $\texttt{MLP}^{(f_{\ell^*})}(\cdot)$.

The low-rank parameters $\{A, B\}$ are then optimized via the Knowledge Seeding objective (defined below) to minimize the distillation loss between teacher and student.

(c) *Update.* After the optimal A, B are found: (a) the base-weight update $\boldsymbol{\theta}^{(f_{\ell^*-1})} \rightarrow \boldsymbol{\theta}_{\text{new}}^{(f_{\ell^*-1})}$ is applied, (b) the low-rank experts previously added to $\texttt{MLP}^{(f_{\ell^*-1})}(\cdot)$ are reset (synaptic pruning), and (c) the new expert $\{A, B\}$ is activated in $\texttt{MLP}^{(f_{\ell^*})}(\cdot)$.

This ordering ensures that the prospective weights $\boldsymbol{\theta}_{\text{new}}^{(f_{\ell^*-1})}$ are available for defining the student model at step 2, while the teacher still faithfully reflects the pre-update state. We model this as a distillation problem: transferring the knowledge stored in the smaller-state teacher $\texttt{LM}_{\boldsymbol{\theta}}$ to the larger-capacity student $\texttt{LM}_{\boldsymbol{\theta}_{\text{exp}}}$.

Knowledge Seeding We present a new form of knowledge transfer, called knowledge seeding, where one or some ***smaller*** models distill their knowledge to a ***larger*** model. This design allows the larger model to preserve existing knowledge in smaller models, while taking advantage of its larger capacity.

Based on the formulation of Knowledge Seeding (KS), we present self-Knowledge Seeding (SKS), where a smaller version of a model (as discussed above) distills the knowledge to a larger version of the model. This distillation process has two critical challenges: (1) Contrary to conventional cases, student has more capacity and so more expressive power than the teacher. Therefore, training the student on the teacher generated dataset (e.g., Kim & Rush (2016)) can result in sub-optimal use of parameters in student model; (2) The model is in sleep stage and so the access to the external information/dataset is limited. Therefore, most popular methods like Hinton et al. (2015) are not applicable. To overcome these challenges, we build upon Generalized Knowledge Distillation (GKD) (Agarwal et al., 2024), which allows a mixture of on-policy student generated data with a teacher-generated data, and present a novel distillation process based on imitation learning.

Rather than replaying raw data, memory consolidation must extract abstractions acquired during the wake phase. Knowledge seeding thus involves two steps: (1) a distillation process where the student receives token-level feedback from teacher logits on self-generated sequences, and (2) an RL-based imitation method that aligns their sampling processes while preserving distilled knowledge.

We start with constructing a dataset $\mathcal{D}$ by sampling from the teacher model, i.e., $\texttt{LM}_{\boldsymbol{\theta}}$. Next, similar to GKD (Agarwal et al., 2024), we define on policy distillation objective as:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{exp}}) = (1-\lambda)\,\mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\mathcal{F}(\texttt{LM}_{\boldsymbol{\theta}} \| \texttt{LM}_{\boldsymbol{\theta}_{\text{exp}}})(y|x)\Big] + \lambda\,\mathbb{E}_{x\sim\mathcal{D}}\Big[\mathbb{E}_{y\sim\texttt{LM}_{\boldsymbol{\theta}_{\text{exp}}}(\cdot|x)}\big[\mathcal{F}(\texttt{LM}_{\boldsymbol{\theta}} \| \texttt{LM}_{\boldsymbol{\theta}_{\text{exp}}})(y|x)\big]\Big]$$

where $\mathcal{F}(\texttt{LM}_{\boldsymbol{\theta}}, \texttt{LM}_{\boldsymbol{\theta}_{\text{exp}}})(y|x)$ is a divergence between teacher (i.e., $\texttt{LM}_{\boldsymbol{\theta}}$) and student (i.e., $\texttt{LM}_{\boldsymbol{\theta}_{\text{exp}}}$) output distributions, and $\lambda \in [0,1]$ controls the fraction of on-policy student-generated outputs. In this optimization process, we freeze all the parameters in the student model and only updates the expanded parameters. This ensures that the transferred knowledge does not interfere with the old knowledge, causing catastrophic forgetting.

Learning to Imitate The above distillation process ensures that the student new parameters store the knowledge encoded in the lower-frequency memory. However, we observe that despite having access to the knowledge, the student model has not learned to use it and so weakly mimics the sampling and performance of the teacher. To this end, we further improve the above distillation process by incorporating RL to teach model how to imitate the teacher sampling. Given a set of teacher generated data (dreams), $\mathcal{D}_T = \{d^{(1)}, \ldots, d^{(n)}\}$, Learning to Imitate (LTI) process first randomly samples a prefix from each $d^{(i)}$ and then asks the student model to complete the continuation. Given the student responses $\hat{d}^{(i)}$ the assigned reward is defined as:

$$r(\hat{d}^{(i)}; d^{(i)}; \texttt{LM}_{\boldsymbol{\theta}_{\text{exp}}}) = \gamma \times r_{\text{sem}}(\hat{d}^{(i)}; d^{(i)}; \texttt{LM}_{\boldsymbol{\theta}_{\text{exp}}}) + (1-\gamma) \times r_{\text{abs}}(\hat{d}^{(i)}; d^{(i)}; \texttt{LM}_{\boldsymbol{\theta}_{\text{exp}}}), \tag{3}$$

where $r_{\text{sem}}(\cdot;\cdot;\cdot)$ (resp. $r_{\text{abs}}(\cdot;\cdot;\cdot)$) assigns a reward based on the semantic similarity (resp. absolute token-level similarity). By incorporating the above LTI process to on-policy distillation, our on-policy knowledge seeding (KS) objective is defined as:

$$\mathcal{L}_{\text{KS}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{exp}}) = \mathbb{E}_{x\sim\mathcal{D}}\Big[(1-\alpha)\mathbb{E}_{y\sim\texttt{LM}_{\boldsymbol{\theta}_{\text{exp}}}(\cdot|x)}\,[r(y)] - \alpha\mathbb{E}_{y\sim\texttt{LM}_{\boldsymbol{\theta}_{\text{exp}}}(\cdot|x)}\big[\mathcal{D}(\texttt{LM}_{\boldsymbol{\theta}} \| \texttt{LM}_{\boldsymbol{\theta}_{\text{exp}}})(y|x)\big]\Big], \tag{4}$$

where $\alpha \in [0,1]$ balances distillation strength against the LTI objective. Optimizing this objective updates expanded parameters and consolidates high-frequency memory into lower-frequency blocks. Once consolidated into $\texttt{MLP}^{(f_{\ell^*})}(\cdot)$, we reset the low-rank parameters previously added to $\texttt{MLP}^{(f_{\ell^*-1})}(\cdot)$, freeing its capacity for future use; a process akin to synaptic pruning in the human brain (Li et al., 2017). For practical implementation details (e.g., masked parameter activation), we refer to Appendix G.

3.4 Dreaming: A Self-Modifying Process

The previous stage, which involved freezing higher-frequency parameters and distilling their knowledge to lower-frequency memories acts similar to slow-wave stage of sleep (NREM) in humans, which is responsible for memory consolidation. In REM stage, however, the brain is highly active (even on par with waking time) and aims to self-modify and strengthen newly formed synapses by dreaming. Inspired by this, we aim to design a dreaming process that learns how to generate dreams (synthetic data) that can help itself to improve over time (Algorithm 5, Appendix C).

In practice, any synthetic data generation process for self-improvement (e.g., Pang et al. (2024a); Huang et al. (2025); Zweiger et al. (2025)) can be incorporated in this stage. A critical consideration,

however, is the risk of iteratively applying self-improvement in continual learning setup, which might cause catastrophic forgetting (Zweiger et al., 2025). In our evaluation, we show that how our two-step design of sleep as memory consolidation and then dreaming as self-modifying process is more robust to catastrophic forgetting. As a proof of concept, we build upon the work of Zweiger et al. (2025), SEAL; however, there are three challenges to incorporate it in our sleep paradigm: (1) Due to the cost of supervised fine-tuning (SFT) in SEAL's inner-loop, it is limited to small number of self-edits (dreams in our terminology). (2) Potential catastrophic forgetting as the cause of iterative self-improvement in sleep periods. (3) The sampling process only samples from the existing knowledge space of the model, while one of the key roles of dreaming is to explore novel synthesis of memories (Stickgold, 2005).

Given a sampled task (C, τ), where C is the context containing information relevant to the task and $\tau(\cdot)$ is a measure to asses the performance in the downstream evaluation, our "dreaming" process starts with generating $m \geq 1$ dreams with having C in context. In the sampling process, each router in MoE blocks additionally chooses a random expert and so incorporates random irrelevant knowledge to the dreaming, learning the underlying patterns that are hidden from model's sight. For this step, we let $\{\texttt{DREAM}^{(i)}\}_{i=1}^{m} \sim \texttt{LM}_{\boldsymbol{\theta}}(\cdot|C)$. Next, we reject some of the generated dreams and only keeps the samples with the most potential in improving the model's performance. To this end, we take inspiration from the literature on gradient-based data selection (Wang et al., 2024a; Pan et al., 2024): for each dream, $\texttt{DREAM}^{(i)}$, we assign an importance score $\boldsymbol{\omega}^{(i)}$ and select $\texttt{Top-}k$ dreams with highest importance score along with b random samples to maintain diversity. Given language modeling objective $\mathcal{L}_{SFT}(\cdot)$, we define importance score of $\texttt{DREAM}^{(i)}$, denoted as $g_{\texttt{DR}}^{(i)}$, as the gradient of the objective: $g_{\texttt{DR}}^{(i)} = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{SFT}(\texttt{DREAM}^{(i)}, \boldsymbol{\theta})$. We let $\texttt{D}$ be the set of all selected dreams by the above process. For each $\texttt{DREAM}^{(i)} \in \texttt{D}$ we consider an isolated instance of the model and updates its parameters via supervised finetuning (with LoRA (Hu et al., 2022)): i.e., $\boldsymbol{\theta}'^{(i)} \leftarrow \texttt{SFT}\left(\boldsymbol{\theta}^{(i)}, \texttt{DREAM}^{(i)}\right)$. Given the new fine-tuned model, following SEAL (Zweiger et al., 2025), we reward the generation of $\texttt{DREAM}^{(i)}$ based on $\texttt{LM}_{\boldsymbol{\theta}'^{(i)}}$'s performance improvement over $\texttt{LM}_{\boldsymbol{\theta}^{(i)}}$:

$$r\left(\texttt{DREAM}^{(i)}, \tau(\cdot), \texttt{LM}_{\boldsymbol{\theta}^{(i)}}\right) = \begin{cases} 1 & \text{If improves,} \\ 0 & \text{Otherwise.} \end{cases} \tag{5}$$

We follow SEAL and use ReST^{EM} algorithm (Singh et al., 2024a) to optimize the above process.

4 Theoretical Analysis

The design choices in the Sleep paradigm—on-policy distillation with reward, low-rank expansion, random expert injection, and gradient filtering—are empirically motivated. In this section, we provide theoretical grounding for these choices. Full formal statements, proofs, and auxiliary lemmas are deferred to Appendix A.

4.1 Analysis of Memory Consolidation

We analyze the Knowledge Seeding (KS) objective used for memory consolidation. Consider a multi-layer network where we are distilling layer $\ell-1$ (sender) into layer ℓ (receiver). Let $h_{\ell-1}$ denote the activations entering the receiver layer, $\Phi_{\text{upper}}(\cdot)$ denote the fixed upper network mapping to logits, and $M = AB^\top$ be the low-rank expansion matrix that the consolidation step aims to learn.

We model the semantic reward component as a projection onto a Semantic Subspace, yielding a combined reward metric $Q = (1-\alpha)\Omega_{\text{sem}} + \alpha\lambda I$, where Ω_{sem} weights semantically critical vocabulary dimensions. We define the *knowledge delta target* $u_{h_{\ell-1}} := Z_T - \Phi_{\text{upper}}(\text{MLP}_\ell(h_{\ell-1}; \theta_\ell))$, which captures the residual knowledge that the base-weight update failed to absorb (see Appendix A.1 for a detailed interpretation).

Under a first-order approximation for the student logits, the KS loss can be written as a quadratic function of the vectorized form of the low-rank matrix, $m = \text{vec}(M)$ (Theorem 4 in Appendix A.1). Specializing to a linear upper network $\Phi_{\text{upper}}(\cdot) = W_u \cdot (\cdot)$ and fully on-policy evaluation ($\lambda = 1$), we obtain the following closed-form solution:

Theorem 1 (Optimal Consolidation as Truncated SVD; informal)**.** *Let $H_S = \mathbb{E}_{h_{\ell-1}^s}[h_{\ell-1}^s (h_{\ell-1}^s)^\top]$ be the on-policy hidden-state covariance and $\tilde{Q} = W_u^\top Q W_u$ the reward metric tensor. Assuming*

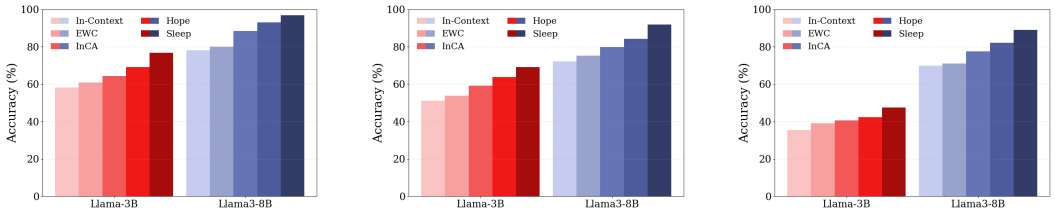


Figure 3: Class-incremental learning on (Left) CLINC (Larson et al., 2019), (Mid.) Banking (Casanueva et al., 2020), (Right) DBpedia (Auer et al., 2007).

W_u is full column rank and H_S is full rank, the rank-r minimizer M_* of the KS loss is

$$M_* = \tilde{Q}^{-1/2}\,\mathrm{SVD}_r\Big(\tilde{Q}^{-1/2}\,C\,H_S^{-1/2}\Big)\;H_S^{-1/2},$$

where $C = W_u^\top Q\,\mathbb{E}_{h_{\ell-1}^s}[u_{\ell-1}^s(h_{\ell-1}^s)^\top]$ and $\mathrm{SVD}_r(\cdot)$ denotes the top-r singular vector projection.

This result reveals two key mechanisms: (i) the on-policy covariance H_S stretches the input space so that the SVD allocates rank budget proportionally to the *student's actual hidden-state distribution*, mitigating exposure bias; and (ii) the reward metric $\tilde{Q}$ dilates the output space along semantically important dimensions, forcing the limited rank budget to prioritize reconstruction accuracy on semantically critical token distinctions. See Appendix A.1 for the formal statement and proof.

4.2 Analysis of the Dreaming Phase

We next examine how random expert injection and gradient-based filtering improve the dreaming phase's self-improvement signal. Let Σ_0 denote the wake-phase gradient covariance.

Theorem 2 (Random Expert Injection Improves Conditioning; informal)**.** *Under the Gauss–Newton approximation, the gradient covariance under random MoE expert injection (with injection strength ε) decomposes as*

$$\Sigma_{\mathrm{raw}} = \Sigma_0 + \varepsilon^2\,\Sigma_{\mathrm{inject}} + O(\varepsilon^3),$$

where $\Sigma_{\mathrm{inject}} = \Sigma_{\mathrm{between}} + \Sigma_{\mathrm{within}}$, with $\Sigma_{\mathrm{between}} \succeq 0$ capturing inter-expert diversity. The injection term Σ_{inject} is confined to the column space of Σ_0 (it cannot expand the rank of the gradient covariance), but provides eigenvalue amplification within the existing gradient subspace, thereby improving the condition number.

We also show that gradient-based filtering preserves the eigenbasis while amplifying all eigenvalues, with amplification ratios that are non-decreasing in eigenvalue magnitude (Theorem 7 in Appendix A.2). Combining injection with the ReSTEM reward step yields:

Theorem 3 (Spectral Whitening via Injection and Reward; informal)**.** *Let μ_1, μ_s be the extreme eigenvalues of Σ_0 on its support $\mathcal{S}$. If the task gradient g_C is aligned with the top eigenvector of $\Sigma_{\mathrm{raw}}|_{\mathcal{S}}$ and a spectral gap separates the dominant eigenvalue ($\lambda_2/\lambda_1 \le 1 - 2/\pi$), then the dreaming phase reduces the condition number to at most*

$$\kappa^+(\Sigma_{\mathrm{dream}}) \le \left(1 - \frac{2}{\pi}\right)\frac{\mu_1 + \varepsilon^2\delta_1}{\mu_s + \varepsilon^2\delta_s} < \left(1 - \frac{2}{\pi}\right)\frac{\mu_1}{\mu_s},$$

where δ_1, δ_s are the extreme eigenvalues of $\Sigma_{\mathrm{inject}}|_{\mathcal{S}}$.

In words, random MoE injection inflates trailing eigenvalues on the support (improving conditioning), while the ReSTEM reward step selectively reduces the top eigenvalue via a rank-1 downdate, providing a multiplicative improvement of $(1 - 2/\pi) \approx 0.36$, in reducing the condition number of the gradient covariance. Full statements and proofs are in Appendix A.2.

5 Empirical Results

We evaluate each sleep stage individually and combined (see Appendix D for additional results.)

5.1 The Effect of Memory Consolidation

We evaluate *only* memory consolidation (without self-improvement) on continual-learning and long-context benchmarks.

Class Incremental Learning We evaluate on CLINC (Larson et al., 2019), Banking (Casanueva et al., 2020), and DBpedia (Auer et al., 2007) using Llama-3B and Llama3-8B (Dubey et al., 2024) as backbones (see Section D.1 for details). Following Momeni et al. (2025), we compare against ICL, EWC (Kirkpatrick et al., 2017), InCA (Momeni et al., 2025), and Hope (Behrouz et al., 2025). Results in Figure 3 show that Hope performs best across all datasets, converting prompt-level adaptation into durable parametric memory.

Effect of Levels (#Sleep Phases) on In-context Learning We evaluate on LongHealth (Adams et al., 2025), QASPER (Dasigi et al., 2021), and MK-NIAH (Hsieh et al., 2024) (see Section D.1), comparing against ICL, DuoAttention (Xiao et al., 2025), and Cartridges (Eyuboglu et al., 2025). Hope consistently outperforms all baselines. Increasing consolidation stages improves performance, while increasing the lowest frequency weakens retention (see Section E.1 for detailed plots).

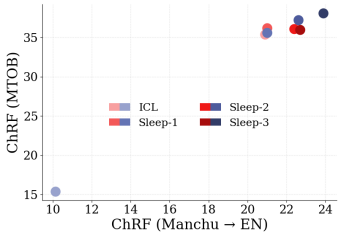


Figure 4: Continual Translation of a Novel Language (CTNL) task. Red: single language; blue: continual learning.

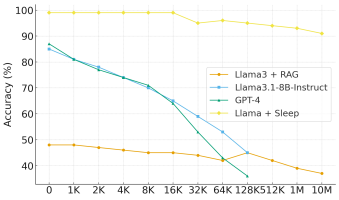


Figure 5: BABILong benchmark. Red: fine-tuned models; blue: zero-shot large models.

Table 1: Ablations on mathematical reasoning (Qwen3-8B, avg@16).

Method	AIME-24	AIME-25	HMMT-25
Sleep	79.2	69.0	46.1
- Imitation Learning	76.8	67.9	45.0
- Semantic Reward	78.9	69.2	44.5
- w/o Expansion	78.2	67.9	44.9
- Reset	78.8	68.8	45.9
- Absolute reward	78.7	68.6	45.5
OPSD	76.6	67.4	45.1
+ Expansion	77.9	68.2	45.9

Learning a New Language In-Context Following Behrouz et al. (2025), we evaluate on MTOB (Tanzer et al., 2024) and Manchu (Pei et al., 2025). Results (Figure 4) show that under continual learning, ICL exhibits a sharp drop, whereas Hope retains its gains, improving with additional consolidation stages.

BABILong On the BABILong benchmark (Kuratov et al., 2024) (Figure 5), Hope achieves near-perfect scores scaling to 10M tokens, outperforming GPT-4/GPT-4o-mini (Achiam et al., 2023) and other baselines (see Section D.2).

Memory Consolidation for Reasoning We evaluate memory consolidation on mathematical reasoning, comparing with base model, SFT, and GRPO (Shao et al., 2024). Results (Table 3 in Section E.2) show our method outperforms all baselines.

Knowledge Incorporation We evaluate Sleep on integrating new facts from SQuAD (Rajpurkar et al., 2016), following Zweiger et al. (2025). Table 4 (in Section E.3) shows Sleep achieves the best results in both settings.

Few-Shot Learning & Ablations Following Akyürek et al. (2024a); Zweiger et al. (2025), Sleep achieves 80% success rate on few-shot ARC (Table 5 in Section E.4), outperforming all baselines. We ablate design choices in Figure 1; all components contribute positively.

6 Conclusion

We introduced Sleep: (i) *knowledge seeding*, an upward distillation transferring short-term knowledge into stable lower-frequency parameters, and (ii) *dreaming*, self-generated training that improves capabilities while controlling interference. KS admits a closed-form SVD solution (Thm. 1), random expert injection improves conditioning (Thm. 2), and the combined pipeline reduces the condition number to $\approx 36\%$ of its original value (Thm. 3). Empirically, Sleep yields consistent gains across continual learning, long-context understanding, knowledge incorporation, and few-shot reasoning.

AI USE STATEMENT

In this work, we used generative AI tools to improve the manuscript's clarity, presentation, and writing, and to reorganize it in accordance with the conference's formatting requirements. Some of the analytical derivations were first obtained by using AI tools. We have not used generative AI tools for conducting experiments. All AI-assisted material was carefully reviewed and verified by the authors. We accept full responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Lisa Adams, Felix Busch, Tianyu Han, Jean-Baptiste Excoffier, Matthieu Ortala, Alexander Löser, Hugo JWL Aerts, Jakob Nikolas Kather, Daniel Truhn, and Keno Bressem. Longhealth: A question answering benchmark with long clinical documents. *Journal of Healthcare Informatics Research*, pp. 1–17, 2025.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. In *Forty-second International Conference on Machine Learning*, 2024a.

Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. In-context language learning: Architectures and algorithms. *arXiv preprint arXiv:2401.12973*, 2024b.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Re. Simple linear attention language models balance the recall-throughput tradeoff. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=e93ffDcpH3.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *international semantic web conference*, pp. 722–735. Springer, 2007.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xlstm: Extended long short-term memory. *arXiv preprint arXiv:2405.04517*, 2024.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*, 2024.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. Nested learning: The illusion of deep learning architectures. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=nbMeRvNb7A.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. Recurrent memory transformer. *Advances in Neural Information Processing Systems*, 35:11079–11091, 2022.

Lucas Caccia, Alan Ansell, Edoardo Ponti, Ivan Vulic, and Alessandro Sordoni. Training plug-n-play knowledge modules with deep context distillation. *arXiv preprint arXiv:2503.08727*, 2025.

Inigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. Efficient intent detection with dual sentence encoders. *ACL 2020*, pp. 38, 2020.

Vivek Chari, Guanghui Qin, and Benjamin Van Durme. Kv-distill: Nearly lossless learnable context compression for llms. *arXiv preprint arXiv:2503.10337*, 2025.

Jeffrey Cheng, Marc Marone, Orion Weller, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. Dated data: Tracing knowledge cutoffs in large language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=wS7PxDjy6m.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. *arXiv preprint arXiv:2305.14788*, 2023.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. *arXiv preprint arXiv:2105.03011*, 2021.

DeepSeek-AI. Deepseek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Benoit Dherin, Michael Munn, Hanna Mazzawi, Michael Wunder, and Javier Gonzalvo. Learning without training: The implicit dynamics of in-context learning. *arXiv preprint arXiv:2507.16003*, 2025.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1107–1128, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.64. URL https://aclanthology.org/2024.emnlp-main.64/.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Will Tennien, Atri Rudra, James Zou, Azalia Mirhoseini, et al. Cartridges: Lightweight and general-purpose long context representations via self-study. *arXiv preprint arXiv:2506.06266*, 2025.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

David J Foster and Matthew A Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440(7084):680–683, 2006.

Uwe Frey and Richard GM Morris. Synaptic tagging and long-term potentiation. *Nature*, 385(6616): 533–536, 1997.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*, 2023.

Andrea N Goldstein and Matthew P Walker. The role of sleep in emotional brain function. *Annual review of clinical psychology*, 10(1):679–708, 2014.

Oscar C. González, Yury Sokolov, Giri P. Krishnan, Jean Erik Delanois, and Maxim Bazhenov. Can sleep protect memories from catastrophic forgetting? *eLife*, 9:e51005, 2020.

Akihiro Goto, Ayaka Bota, Ken Miya, Jingbo Wang, Suzune Tsukamoto, Xinzhi Jiang, Daichi Hirai, Masanori Murayama, Tomoki Matsuda, Thomas J. McHugh, Takeharu Nagai, and Yasunori Hayashi. Stepwise synaptic plasticity events drive the early phase of memory consolidation. *Science*, 374(6569):857–863, 2021. doi: 10.1126/science.abj9195. URL https://www.science.org/doi/abs/10.1126/science.abj9195.

Yunhao Gou, Zhili Liu, Kai Chen, Lanqing Hong, Hang Xu, Aoxue Li, Dit-Yan Yeung, James T Kwok, and Yu Zhang. Mixture of cluster-conditional lora experts for vision-language instruction tuning. *arXiv preprint arXiv:2312.12379*, 2023.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 2451–2463, 2018a.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018b.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. Self-distillation zero: Self-revision turns binary rewards into dense supervision, 2026. URL https://arxiv.org/abs/2604.12002.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Audrey Huang, Adam Block, Dylan J Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T. Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=WJaUkwci9o.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*, 2023.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats Leon Richter, Quentin Gregory Anthony, Eugene Belilovsky, Timothée Lesort, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=DimPeeCxKO.

Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. A modern self-referential weight matrix that learns to modify itself. In *International Conference on Machine Learning*. PMLR, 2022. URL https://proceedings.mlr.press/v162/irie22b.html.

Daoyun Ji and Matthew A Wilson. Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature neuroscience*, 10(1):100–107, 2007.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*, 2023.

Michael V Johnston. Plasticity in the developing brain: implications for rehabilitation. *Developmental disabilities research reviews*, 15(2):94–101, 2009.

Eric R Kandell, Johjn D Koester, Sarah H Mack, and Steven Siegelbaum. *Principles of neural science*. McGraw-Hill, 2021.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of llms?, 2026. URL https://arxiv.org/abs/2603.24472.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

John Kirchenbauer, Abhimanyu Hans, Brian Bartoldson, Micah Goldblum, Ashwinee Panda, and Tom Goldstein. Multi-token prediction via self-distillation, 2026. URL https://arxiv.org/abs/2602.06019.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Kalle Kujanpää, Harri Valpola, and Alexander Ilin. Knowledge injection via prompt distillation. *arXiv preprint arXiv:2412.14964*, 2024.

Dharshan Kumaran, Demis Hassabis, and James L. McClelland. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in Cognitive Sciences*, 20 (7):512–534, 2016.

Yury Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *Advances in Neural Information Processing Systems*, 37:106519–106554, 2024.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1311–1316, 2019.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, et al. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts. *arXiv preprint arXiv:2404.15159*, 2024a.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context LLMs struggle with long in-context learning. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Cw2xlg0e46.

Wei Li, Lei Ma, Guang Yang, and Wen-Biao Gan. Rem sleep selectively prunes and maintains new synapses in development and learning. *Nature neuroscience*, 20(3):427–437, 2017.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

Yucheng Li. Unlocking context constraints of llms: Enhancing context efficiency of llms with self-information-based content filtering. *arXiv preprint arXiv:2304.12102*, 2023.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024b.

Kevin Lin, Charlie Snell, Yu Wang, Charles Packer, Sarah Wooders, Ion Stoica, and Joseph E Gonzalez. Sleep-time compute: Beyond inference scaling at test-time. *arXiv preprint arXiv:2504.13171*, 2025.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3–4):293–321, 1992.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. Minicache: Kv cache compression in depth dimension for large language models. *Advances in Neural Information Processing Systems*, 37, 2024.

Yihong Liu, Raoyuan Zhao, Michael A. Hedderich, and Hinrich Schütze. Crosslingual on-policy self-distillation for multilingual reasoning, 2026. URL https://arxiv.org/abs/2605.09548.

Yansheng Mao, Yufei Xu, Jiaqi Li, Fanxu Meng, Haotong Yang, Zilong Zheng, Xiyuan Wang, and Muhan Zhang. Lift: Improving long context understanding of large language models through long input fine-tuning. *arXiv preprint arXiv:2502.14644*, 2025.

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.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37: 121038–121072, 2024.

Leon Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 1960.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Saleh Momeni, Sahisnu Mazumder, Zixuan Ke, and Bing Liu. In-context continual learning assisted by an external continual learner. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 7292–7306, 2025.

Nihal V Nayak, Yiyang Nan, Avi Trost, and Stephen H Bach. Learning to generate instruction tuning datasets for zero-shot task adaptation. *arXiv preprint arXiv:2402.18334*, 2024.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. Codegen: An open large language model for code with multi-turn program synthesis. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=iaYcJKpY2B_.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Xingyuan Pan, Luyang Huang, Liyan Kang, Zhicheng Liu, Yu Lu, and Shanbo Cheng. G-DIG: Towards gradient-based DIverse and hiGh-quality instruction data selection for machine translation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15395–15406, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.821. URL https://aclanthology.org/2024.acl-long.821/.

Jing-Cheng Pang, Pengyuan Wang, Kaiyuan Li, Xiong-Hui Chen, Jiacheng Xu, Zongzhang Zhang, and Yang Yu. Language model self-improvement by reinforcement learning contemplation. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=38E4yUbrgr.

Jing-Cheng Pang, Pengyuan Wang, Kaiyuan Li, Xiong-Hui Chen, Jiacheng Xu, Zongzhang Zhang, and Yang Yu. Language model self-improvement by reinforcement learning contemplation. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=38E4yUbrgr.

Alvaro Pascual-Leone, Amir Amedi, Felipe Fregni, and Lotfi B Merabet. The plastic human brain cortex. *Annu. Rev. Neurosci.*, 28(1):377–401, 2005.

Renhao Pei, Yihong Liu, Peiqin Lin, François Yvon, and Hinrich Schütze. Understanding in-context machine translation for low-resource languages: A case study on manchu. *arXiv preprint arXiv:2502.11862*, 2025.

Adrien Peyrache, Mehdi Khamassi, Karim Benchenane, Sidney I Wiener, and Francesco P Battaglia. Replay of rule-learning related neural patterns in the prefrontal cortex during sleep. *Nature neuroscience*, 12(7):919–926, 2009.

Ruiyang Qin, Qingzhuo Wang, Dongrui Liu, Qiang Li, Zhihua Wei, and Wen Shen. Multilingual safety alignment via self-distillation, 2026. URL https://arxiv.org/abs/2605.02971.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016. URL https://aclanthology.org/D16-1264/.

Björn Rasch and Jan Born. About sleep's role in memory. *Physiological reviews*, 2013.

Haris Riaz, Sourav Bhabesh, Vinayak Arannil, Miguel Ballesteros, and Graham Horwood. Metasynth: Meta-prompting-driven agentic scaffolds for diverse synthetic data generation. *arXiv preprint arXiv:2504.12563*, 2025.

Ivan Rodkin, Yuri Kuratov, Aydar Bulatov, and Mikhail Burtsev. Associative recurrent memory transformer. *arXiv preprint arXiv:2407.04841*, 2024.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. Crisp: Compressed reasoning via iterative self-policy distillation, 2026. URL https://arxiv.org/abs/2603.05433.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? *Advances in neural information processing systems*, 36:55565–55581, 2023.

Juergen Schmidhuber. Learning to control fast-weight memories: An alternative to recurrent nets. accepted for publication in. *Neural Computation*, 1992.

Jürgen Schmidhuber. Evolutionary principles in self-referential learning, 1987. URL https://people.idsia.ch/~juergen/diploma1987ocr.pdf.

William Beecher Scoville and Brenda Milner. Loss of recent memory after bilateral hippocampal lesions. *Journal of neurology, neurosurgery, and psychiatry*, 20(1):11, 1957.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026a.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026b. URL https://arxiv.org/abs/2601.19897.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 2024.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T Parisi, Abhishek Kumar, Alexander A Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura A Culp, Lechao Xiao, Maxwell Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856. URL https://openreview.net/forum?id=lNAyUngGFK. Expert Certification.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T Parisi, Abhishek Kumar, Alexander A Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura A Culp, Lechao Xiao, Maxwell Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024b. URL https://openreview.net/forum?id=lNAyUngGFK.

Larry R Squire and Pablo Alvarez. Retrograde amnesia and memory consolidation: a neurobiological perspective. *Current opinion in neurobiology*, 5(2):169–177, 1995.

Larry R Squire, Lisa Genzel, John T Wixted, and Richard G Morris. Memory consolidation. *Cold Spring Harbor perspectives in biology*, 7(8):a021766, 2015.

Alex Stein, Furong Huang, and Tom Goldstein. Gates: Self-distillation under privileged context with consensus gating, 2026. URL https://arxiv.org/abs/2602.20574.

Robert Stickgold. Sleep-dependent memory consolidation. *Nature*, 437(7063):1272–1278, 2005.

Weihang Su, Yichen Tang, Qingyao Ai, Junxi Yan, Changyue Wang, Hongning Wang, Ziyi Ye, Yujia Zhou, and Yiqun Liu. Parametric retrieval augmented generation. *arXiv preprint arXiv:2501.15915*, 2025.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

Timothy Tadros, Giri P. Krishnan, Ramyaa Ramyaa, and Maxim Bazhenov. Sleep-like unsupervised replay reduces catastrophic forgetting in artificial neural networks. *Nature Communications*, 13(1): 7742, 2022.

Sijun Tan, Xiuyu Li, Shishir Patil, Ziyang Wu, Tianjun Zhang, Kurt Keutzer, Joseph E Gonzalez, and Raluca Ada Popa. Lloco: Learning long contexts offline. *arXiv preprint arXiv:2404.07979*, 2024.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. *arXiv preprint arXiv:2406.10774*, 2024.

Garrett Tanzer, Mirac Suzgun, Eline Visser, Dan Jurafsky, and Luke Melas-Kyriazi. A benchmark for learning to translate a new language from one grammar book. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tbVWug9f2h.

Giulio Tononi and Chiara Cirelli. Sleep function and synaptic homeostasis. *Sleep medicine reviews*, 10(1):49–62, 2006.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Erin J Wamsley and Robert Stickgold. Memory, sleep and dreaming: experiencing consolidation. *Sleep medicine clinics*, 6(1):97, 2011.

Zhongwei Wan, Xinjian Wu, Yu Zhang, Yi Xin, Chaofan Tao, Zhihong Zhu, Xin Wang, Siqi Luo, Jing Xiong, and Mi Zhang. D2o: Dynamic discriminative operations for efficient generative inference of large language models. *arXiv preprint arXiv:2406.13035*, 2024.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. Skill-sd: Skill-conditioned self-distillation for multi-turn llm agents, 2026. URL https://arxiv.org/abs/2604.10674.

Jiachen Tianhao Wang, Tong Wu, Dawn Song, Prateek Mittal, and Ruoxi Jia. Greats: Online selection of high-quality data for llm training in every iteration. *Advances in Neural Information Processing Systems*, 37:131197–131223, 2024a.

Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *Advances in Neural Information Processing Systems*, 36:61501–61513, 2023.

Zhaoyang Wang, Weilei He, Zhiyuan Liang, Xuchao Zhang, Chetan Bansal, Ying Wei, Weitong Zhang, and Huaxiu Yao. CREAM: Consistency regularized self-rewarding language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Vf6RDObyEF.

Zheng Wang, Boxiao Jin, Zhongzhi Yu, and Minjia Zhang. Model tells you where to merge: Adaptive kv cache merging for llms on long-context tasks. *arXiv preprint arXiv:2407.08454*, 2024b.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of lora experts. *arXiv preprint arXiv:2404.13628*, 2024.

Chaojun Xiao, Zhengyan Zhang, Chenyang Song, Dazhi Jiang, Feng Yao, Xu Han, Xiaozhi Wang, Shuo Wang, Yufei Huang, Guanyu Lin, et al. Configurable foundation models: Building llms from a modular perspective. *arXiv preprint arXiv:2409.02877*, 2024.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, junxian guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. Duoattention: Efficient long-context LLM inference with retrieval and streaming heads. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=cFu7ze7xUm.

Prateek Yadav, Colin Raffel, Mohammed Muqeeth, Lucas Caccia, Haokun Liu, Tianlong Chen, Mohit Bansal, Leshem Choshen, and Alessandro Sordoni. A survey on model moerging: Recycling and routing among specialized experts for collaborative learning. *arXiv preprint arXiv:2408.07057*, 2024.

Wannan Yang, Chen Sun, Roman Huszár, Thomas Hainmueller, Kirill Kiselev, and György Buzsáki. Selection of experience for memory by hippocampal sharp wave ripples. *Science*, 383(6690): 1478–1483, 2024.

Tianzhu Ye, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Online experiential learning for language models, 2026a. URL https://arxiv.org/abs/2603.16856.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models, 2026b. URL https://arxiv.org/abs/2602.12275.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping reasoning with reasoning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/639a9a172c044fbb64175b5fad42e9a5-Paper-Conference.pdf.

Rongzhi Zhang, Kuang Wang, Liyuan Liu, Shuohang Wang, Hao Cheng, Chao Zhang, and Yelong Shen. Lorc: Low-rank compression for llms kv cache with a progressive compression strategy. *arXiv preprint arXiv:2410.03111*, 2024a.

Ruixiang Zhang, Richard He Bai, Huangjie Zheng, Navdeep Jaitly, Ronan Collobert, and Yizhe Zhang. Embarrassingly simple self-distillation improves code generation, 2026a. URL https://arxiv.org/abs/2604.01193.

Xinsen Zhang, Zhenkai Ding, Tianjun Pan, Run Yang, Chun Kang, Xue Xiong, and Jingnan Gu. Opsdl: On-policy self-distillation for long-context language models. *arXiv preprint arXiv:2604.17535*, 2026b.

Xinsen Zhang, Zhenkai Ding, Tianjun Pan, Run Yang, Chun Kang, Xue Xiong, and Jingnan Gu. Opsdl: On-policy self-distillation for long-context language models, 2026c. URL https://arxiv.org/abs/2604.17535.

Yaocheng Zhang, Yuanheng Zhu, Wenyue Chong, Songjun Tu, Qichao Zhang, Jiajun Chai, Xiaohan Wang, Wei Lin, Guojun Yin, and Dongbin Zhao. π-play: Multi-agent self-play via privileged self-distillation without external data, 2026d. URL https://arxiv.org/abs/2604.14054.

Yuxin Zhang, Yuxuan Du, Gen Luo, Yunshan Zhong, Zhenyu Zhang, Shiwei Liu, and Rongrong Ji. Cam: Cache merging for memory-efficient llms inference. In *Forty-first International Conference on Machine Learning*, 2024b.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026a.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models, 2026b. URL https://arxiv.org/abs/2601.18734.

Ziyu Zhao, Leilei Gan, Guoyin Wang, Wangchunshu Zhou, Hongxia Yang, Kun Kuang, and Fei Wu. Loraretriever: Input-aware lora retrieval and composition for mixed tasks in the wild. *arXiv preprint arXiv:2402.09997*, 2024a.

Ziyu Zhao, Tao Shen, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, Kun Kuang, and Fei Wu. Merging loras like playing lego: Pushing the modularity of lora to extremes through rank-wise clustering. *arXiv preprint arXiv:2409.16167*, 2024b.

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

A Theoretical Analysis

This appendix provides the complete theoretical foundation, formal mathematical statements, extended discussions, and full proofs underlying the theoretical analysis presented in Section 4. We begin with the formal framework for the Memory Consolidation (Knowledge Seeding) phase in Section A.1, followed by the formal analysis of the Dreaming phase in Section A.2, auxiliary results connecting logit distances to KL divergence in Section A.3, and complete step-by-step proofs for all theoretical results in Section A.4 and Section A.5.

A.1 Analysis of Memory Consolidation

In this section, we analyze the Knowledge Seeding (KS) objective used during the memory consolidation stage. We consider a multi-layer transformer network containing J layers, and suppose that the sleep cycle is consolidating knowledge from layer $\ell - 1$ (the faster-updating sender layer) into layer ℓ (the slower-updating receiver layer).

Let h_{in} denote the hidden activations entering layer $\ell - 1$. Let $\Phi_{\text{upper}}(\cdot) = W_{\text{vocab}} \circ \text{MLP}_J \circ \cdots \circ \text{MLP}_{\ell+1}(\cdot)$ denote the fixed mapping through the upper layers and the final vocabulary projection head.

Generalized Knowledge Distillation (GKD). Following Agarwal et al. (2024), Generalized Knowledge Distillation blends off-policy data distributions with on-policy trajectories generated by the student:

$$\begin{aligned}\mathcal{L}_{\text{GKD}}(\mathcal{M}_T, \mathcal{M}_S) = (1-\lambda)\,\mathbb{E}_{(x_0,y_{1:T})\sim\mathcal{D}}\big[D_{\text{KL}}(\mathcal{M}_T\|\mathcal{M}_S)(x_0, y_{1:T})\big] \\ + \lambda\,\mathbb{E}_{x_0\sim\mathcal{D}}\big[\mathbb{E}_{y_{1:T}\sim\mathcal{M}_S}[D_{\text{KL}}(\mathcal{M}_T\|\mathcal{M}_S)(x_0, y_{1:T})]\big],\end{aligned} \tag{6}$$

where $\lambda \in [0, 1]$ parameterizes the fraction of on-policy student generation, and D_{KL} is the token-level KL divergence $D_{\text{KL}}(\mathcal{M}_T\|\mathcal{M}_S) = \sum_{v\in\mathcal{V}} \mathcal{M}_T(v|x) \log \frac{\mathcal{M}_T(v|x)}{\mathcal{M}_S(v|x)}$.

Teacher and Student Logit Formulations. During memory consolidation:

- **Teacher Logits (Z_T):** The teacher evaluates the model prior to the base-weight update, utilizing the accumulated experts $E_{\ell-1}$ stored in the fast layer $\ell - 1$:

$$Z_T = \Phi_{\text{upper}}\Big(\text{MLP}_\ell\big(\text{MLP}_{\ell-1}(h_{\text{in}}; \theta^{\text{old}}_{\ell-1} + E_{\ell-1}); \theta_\ell\big)\Big). \tag{7}$$

- **Student Logits (Z_S):** The student evaluates the model using the prospective updated base weights $\theta^{\text{new}}_{\ell-1}$ (with experts in layer $\ell - 1$ reset for subsequent wake cycles), routing activations through layer ℓ expanded by the newly injected low-rank matrices $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{d\times r}$ with $M = AB^\top$:

$$Z_S = \Phi_{\text{upper}}\Big(\text{MLP}_\ell\big(\text{MLP}_{\ell-1}(h_{\text{in}}; \theta^{\text{new}}_{\ell-1}); \theta_\ell\big) + AB^\top \text{MLP}_{\ell-1}(h_{\text{in}}; \theta^{\text{new}}_{\ell-1})\Big). \tag{8}$$

Because the low-rank matrices A and B operate within the network at layer ℓ, the logit-level Knowledge Seeding (KS) loss must be analytically projected back through the highly nonlinear upper layers Φ_{upper}. Here, we are assuming a linear activation for the added expert for simplicity. The notation $\theta + E$ serves as an algebraic abstraction for parallel parameter expansion. In Mixture of Experts (MoE) or Low-Rank Adaptation (LoRA) architectures, the low-rank expert matrix $E = AB^T$ bypasses the strict sequential bottleneck by acting as an additive linear bypass alongside the frozen base projections. Because linear transformations distribute over addition, injecting the expert parallel to the base weights structurally equates to evaluating the operation under the superposed parameter tensor $\theta + E$.

We adopt the shorthand $h_{\ell-1} := \text{MLP}_{\ell-1}(h_{\text{in}}; \theta^{\text{new}}_{\ell-1})$. By Theorem 11 (proved in Section A.3), the KL divergence between the resulting softmax distributions is upper bounded by the squared ℓ_2 distance of their logits: $D_{\text{KL}}(P_T\|P_S) \le \frac{1}{4}\|Z_T - Z_S\|^2$. Minimizing the squared logit distance thus conservatively minimizes the true KL divergence.

While we presented the GKD loss in equation 6, the sleep phase occurs at a specific time t. During this time, the parameter matrices (including the low-rank expansions A and B) are shared identically

across all token positions during a single forward pass. Fixing t, we define $x = (x_0, \ldots, x_{t-1})$, and let $y \in \mathbb{R}^{L\times d}$ denote the next input. The choice of y (either from data or student's on-policy) would be reflected in $h_{\ell-1}$. We let $h^d_{\ell-1}$ be the output of layer $\ell-1$ (the slow layer in distillation) when y is from data and $h^s_{\ell-1}$ is defined likewise when y is generated from student's policy. Therefore, $h^d_{\ell-1}$ is deterministic, while $h^s_{\ell-1}$ is random. The GKD loss can be written as

$$\mathcal{L}_{\mathrm{GKD}}(\mathcal{M}_T, \mathcal{M}_S) \approx \frac{1}{2}(1-\lambda)\left\|\Phi_{upper}\big(\mathrm{MLP}_\ell(h^d_{\ell-1};\theta_\ell) + AB^T h^d_{\ell-1}\big) - Z_T\right\|^2 + \frac{1}{2}\lambda\mathbb{E}_{h^s_{\ell-1}}\left\|\Phi_{upper}\big(\mathrm{MLP}_\ell(h^s_{\ell-1};\theta_\ell) + AB^T h^s_{\ell-1}\big) - Z_T\right\|^2 \tag{9}$$

We next proceed with characterizing the knowledge seeding (KS) objective, given by equation 4. As discussed in the main text, the KS objective uses Levenshtein distance and Semantic similarity for the reward. This indicates that within the continuous logit space, the objective function does not penalize deviations in the exact probability values, as long as the top-tier semantic token predictions remain accurate. We model this mathematically as a projection onto a Semantic Subspace Ω_{sem} (a positive semi-definite matrix that weights semantically critical vocabulary dimensions heavily, and irrelevant dimensions at zero). The reward maximizes the match in this subspace:

$$-r(y) \propto \mathbb{E}_{h^s_{\ell-1}}\|Z_S - Z_T\|^2_{\Omega_{\mathrm{sem}}}$$

Note that the reward function is evaluated on the student's on-policy.

The KS Objective: Substituting these into the loss function (and scaling by a factor 2), we get:

$$\mathcal{L}_{\mathrm{KS}}(\mathcal{M}_T, \mathcal{M}_S) \approx (1-\alpha)\mathbb{E}_{h^s_{\ell-1}}\|Z_S(h^s_{\ell-1}) - Z_T\|^2_{\Omega_{\mathrm{sem}}} + \alpha(1-\lambda)\|Z_S(h^d_{\ell-1}) - Z_T\|^2 + \alpha\lambda\mathbb{E}_{h^s_{\ell-1}}\|Z_S(h^s_{\ell-1}) - Z_T\|^2$$

where we explicitly write the dependence of student's logit on the input to layer $\ell-1$. Define $Q := (1-\alpha)\Omega_{\mathrm{sem}} + \alpha\lambda I$, which acts as the reward matrix. The objective can then be written as

$$\mathcal{L}_{\mathrm{KS}}(\mathcal{M}_T, \mathcal{M}_S) \approx \alpha(1-\lambda)\|Z_S(h^d_{\ell-1}) - Z_T\|^2 + \mathbb{E}_{h^s_{\ell-1}}\|Z_S(h^s_{\ell-1}) - Z_T\|^2_Q$$

where for a vector x, the Mahalanobis norm is defined as $\|x\|^2_Q = x^T Q x$ for a positive semi-definite matrix $Q \succeq 0$. Since $\Omega_{\mathrm{sem}} \succeq 0$, we also have $Q \succeq 0$.

First Order Approximation. Note that in consolidation phase, only the low rank parts A, B are optimized to minimize the KS loss and the loss depends on them through the term $M = AB^T$. With slight abuse of notation, we write $\mathcal{L}(M)$ to highlight this dependence. Using a first-order Taylor expansion of the student logits with respect to the adapter output:

$$Z_S(h_{\ell-1}) := \Phi_{\mathrm{upper}}\big(\mathrm{MLP}_\ell(h_{\ell-1};\theta_\ell) + AB^T h_{\ell-1}\big) \approx \Phi_{\mathrm{upper}}\big(\mathrm{MLP}_\ell(h_{\ell-1};\theta_\ell)\big) + J_{h_{\ell-1}} M h_{\ell-1} \tag{10}$$

where $J_{h_{\ell-1}} := \nabla_x \Phi_{\mathrm{upper}}(x)\big|_{x=\mathrm{MLP}_\ell(h_{\ell-1};\theta_\ell)} \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the Jacobian of the upper network.

Knowledge Delta Target. We define the knowledge delta target vector:

$$u_{h_{\ell-1}} := Z_T - \Phi_{\mathrm{upper}}\big(\mathrm{MLP}_\ell(h_{\ell-1};\theta_\ell)\big). \tag{11}$$

Expanding this definition reveals its core physical meaning:

$$u_{h_{\ell-1}} = \Big(\Phi_{\mathrm{upper}}\big(\mathrm{MLP}_\ell(\mathrm{MLP}_{\ell-1}(h_{\mathrm{in}};\theta^{\mathrm{old}}_{\ell-1} + E_{\ell-1});\theta_\ell)\big) - \Phi_{\mathrm{upper}}\big(\mathrm{MLP}_\ell(\mathrm{MLP}_{\ell-1}(h_{\mathrm{in}};\theta^{\mathrm{old}}_{\ell-1});\theta_\ell)\big)\Big) + \Big(\Phi_{\mathrm{upper}}\big(\mathrm{MLP}_\ell(\mathrm{MLP}_{\ell-1}(h_{\mathrm{in}};\theta^{\mathrm{old}}_{\ell-1});\theta_\ell)\big) - \Phi_{\mathrm{upper}}\big(\mathrm{MLP}_\ell(\mathrm{MLP}_{\ell-1}(h_{\mathrm{in}};\theta^{\mathrm{new}}_{\ell-1});\theta_\ell)\big)\Big). \tag{12}$$

The first bracketed difference represents the temporary short-term knowledge accumulated during the wake cycle in $E_{\ell-1}$. The second bracketed difference represents the portion of knowledge absorbed into the base weights during the base update ($\theta^{\mathrm{old}}_{\ell-1} \to \theta^{\mathrm{new}}_{\ell-1}$). If the base update perfectly captures the wake-phase experience, these terms cancel and $u_{h_{\ell-1}} = 0$. Hence, $u_{h_{\ell-1}}$ represents the *residual*

knowledge that the base update failed to absorb, which the receiver low-rank adapter $M = AB^\top$ is tasked with capturing.

Using the first-order approximation equation 10, and the notation of the knowledge delta target $u_{h_{\ell-1}}$, the Knowledge Seeding loss becomes:

$$\mathcal{L}_{\mathrm{KS}}(M) \approx \alpha(1-\lambda)\left\|J_{h^d_{\ell-1}} M h^d_{\ell-1} - u_{h^d_{\ell-1}}\right\|^2 + \mathbb{E}_{h^s_{\ell-1}}\left[\left\|J_{h^s_{\ell-1}} M h^s_{\ell-1} - u_{h^s_{\ell-1}}\right\|_Q^2\right]. \tag{13}$$

Theorem 4 (Quadratic Form of Knowledge Seeding Loss). *Let $m = \mathrm{vec}(M) \in \mathbb{R}^{d^2}$. The approximated Knowledge Seeding loss $\mathcal{L}_{\mathrm{KS}}(M)$ in equation 13 is a quadratic program in m:*

$$\tilde{\mathcal{L}}(m) = m^\top H m - 2c^\top m + \mathrm{const}, \tag{14}$$

where $H \in \mathbb{R}^{d^2\times d^2}$ and $c \in \mathbb{R}^{d^2}$ are given by:

$$H = \alpha(1-\lambda)\left((h^d_{\ell-1}(h^d_{\ell-1})^\top) \otimes (J^\top_{h^d_{\ell-1}} J_{h^d_{\ell-1}})\right) + \mathbb{E}_{h^s_{\ell-1}}\left[(h^s_{\ell-1}(h^s_{\ell-1})^\top) \otimes (J^\top_{h^s_{\ell-1}} Q J_{h^s_{\ell-1}})\right], \tag{15}$$

$$c = \alpha(1-\lambda)\,\mathrm{vec}\big(J^\top_{h^d_{\ell-1}} u_{h^d_{\ell-1}} (h^d_{\ell-1})^\top\big) + \mathbb{E}_{h^s_{\ell-1}}\left[\mathrm{vec}\big(J^\top_{h^s_{\ell-1}} Q u_{h^s_{\ell-1}} (h^s_{\ell-1})^\top\big)\right]. \tag{16}$$

While the unconstrained minimizer of $\tilde{\mathcal{L}}(m)$ has the closed form solution $m^* = H^\dagger c$, and $M^* = \mathrm{reshape}(M, d, d)$ the rank constrained optimization does not have a closed form solution.

Linear Upper Network and Exact Closed-Form SVD Solution. As a special case, we consider a network where Φ_{upper} is a linear function given by $\Phi_{\mathrm{upper}}(\cdot) = W_u \times (\cdot)$. We also assume a completely on-policy strategy,i.e. $\lambda = 1$. Note that the two layers ($\ell - 1$ and ℓ) that are under the sleep phase and all the lower layers can be non-linear.

In this case, the first order approximation to Z_S in equation 10 becomes exact. Further the KS loss in equation 13 simplifies to:

$$\mathcal{L}_{\mathrm{KS}}(M) = \mathbb{E}_{h^s_{\ell-1}}\left[\left\|W_u M h^s_{\ell-1} - u_{h^s_{\ell-1}}\right\|_Q^2\right]. \tag{17}$$

The following theorem establishes a closed-form solution for the rank-r minimizer of this loss.

Theorem 5 (SVD Closed-Form of Knowledge Delta). *Define $H_S := \mathbb{E}_{h^s_{\ell-1}}[h^s_{\ell-1}(h^s_{\ell-1})^\top] \in \mathbb{R}^{d\times d}$ as the second-moment matrix of activations entering layer ℓ under the student's on-policy trajectory, and let $\tilde{Q} := W_u^\top Q W_u \in \mathbb{R}^{d\times d}$. Define the cross-covariance target $C := W_u^\top Q\, \mathbb{E}_{h^s_{\ell-1}}[u_{h^s_{\ell-1}}(h^s_{\ell-1})^\top]$. Assuming W_u is full column rank and H_S is full rank, the rank-r global minimizer $M_* = \arg\min_{\mathrm{rank}(M)\le r} \mathcal{L}_{\mathrm{KS}}(M)$ of the Knowledge Seeding loss equation 17 is uniquely given by:*

$$M_* = \tilde{Q}^{-1/2}\,\mathrm{SVD}_r\left(\tilde{Q}^{-1/2}\, C\, H_S^{-1/2}\right) H_S^{-1/2}, \tag{18}$$

where $\mathrm{SVD}_r(X)$ denotes the best rank-r Frobenius approximation of X via its top r singular components.

Discussion and Methodological Remarks.

Remark 1 (Interpretation of the Knowledge Delta as Residual). *Equation (12) clarifies that $u_{h_{\ell-1}}$ represents residual knowledge. When the fast layer's updates are successfully absorbed into base parameters, $u_{h_{\ell-1}} \to 0$, obviating redundant storage. The slow layer's adapter $AB^\top$ is allocated exclusively to unabsorbed residual information.*

Remark 2 (Role of the Upper Network as a Downstream Task Lens). *The matrix W_u (or the Jacobian $J_{h_{\ell-1}}$ in the general non-linear case) acts as a task-relevant lens: it projects residual activation errors into logit errors, weighting residual directions by their downstream impact on task loss. Directions that are amplified by the upper network receive larger effective weight in the SVD, ensuring the limited rank budget is allocated to the most consequential components.*

Remark 3 (Role of On-Policy Covariance $H_S^{1/2}$: Mitigating Exposure Bias). *The right factor $H_S^{-1/2}$ in equation 18 whitens the input distribution so that the SVD allocates capacity proportionally to*

the student's actual activation distribution. If distillation were performed off-policy on teacher trajectories, the low-rank bottleneck would overfit to the teacher's exact states. However, the Student has limited capacity rank(r) and will follow a different trajectory. By pulling the SVD through $H_S^{1/2}$, the algorithm stretches the principal components to cover the Student's actual search space. This is the theoretical mechanism by which on-policy evaluation mitigates exposure bias *in the consolidation step.*

Remark 4 (Role of Reward Metric Tensor Q: Semantic Prioritization)**.** *Standard KL distillation treats all token discrepancies equally, often spending low-rank capacity matching syntactic nuances (e.g., "the" vs. "a"). Setting $\alpha < 1$ with semantic reward matrix Ω_{sem} dilates semantically critical dimensions through $\tilde{Q} = W_u^\top Q W_u$, ensuring the truncated SVD prioritizes logical and factual correctness.*

Remark 5 (Role of Parameter Expansion vs. Overwriting)**.** *Directly updating existing weights overwrites previously consolidated knowledge, triggering catastrophic forgetting. Expanding the parameter set by adding a new low-rank expert $\{A, B\}$ provides isolated capacity for the new knowledge, ensuring strict non-interference with previously stored representations.*

Remark 6 (Role of Synaptic Pruning)**.** *Once knowledge is consolidated into the slower receiver layer ℓ, the temporary experts in the fast sender layer $\ell - 1$ are reset. This biological-inspired pruning frees capacity in the fast module for future wake-phase adaptation.*

A.2 Analysis of the Dreaming Phase

We now present the formal analysis of the dreaming phase, establishing how random expert injection, gradient-based filtering, and ReSTEM reward combine to produce spectral whitening of the parameter gradient covariance.

Architecture and Expert Routing. Let $\Phi(x;\theta) = W_{\text{vocab}} \circ \text{MLP}_L \circ \cdots \circ \text{MLP}_1(x)$ be the language model. At layer ℓ, standard deterministic routing activates a top-K subset $\mathcal{K}_{\text{det}} \subset [E]$ of experts:

$$h_\ell^{\text{base}} = \sum_{k \in \mathcal{K}_{\text{det}}} g_k \, \text{MLP}_k^{(\ell)}(h_{\ell-1}). \tag{19}$$

During dreaming, random expert injection perturbs this activation with injection strength $\varepsilon \in (0, 1)$:

$$\tilde{h}_\ell = (1-\varepsilon)\, h_\ell^{\text{base}} + \varepsilon \, \text{MLP}_r^{(\ell)}(h_{\ell-1}) = h_\ell^{\text{base}} + \varepsilon \, \Delta h_r, \qquad r \sim \text{Uniform}\big([E] \setminus \mathcal{K}_{\text{det}}\big), \tag{20}$$

where $\Delta h_r = \text{MLP}_r^{(\ell)}(h_{\ell-1}) - h_\ell^{\text{base}}$ is the expert displacement vector.

Loss, Predictive Distribution, and Gradients. For input context x and injected expert r, the model outputs logits $Z_r = \Phi_{\text{upper}}(\tilde{h}_\ell)$. The SFT loss with respect to a target token y sampled from the temperature-T distribution $P_{T|r} = \text{softmax}(Z_r/T)$ is given by $\mathcal{L}(Z_r, y) = -\log P_{1|r}(y)$, where $P_{1|r} = \text{softmax}(Z_r)$ is the model's predictive distribution at temperature 1. Concretely,

$$\mathcal{L}(Z, y) = -\log\left(\frac{\exp(z_y)}{\sum_k \exp(z_k)}\right) = -z^T e_y + \log \sum_{k=1}^{|V|} \exp(z_k)$$

The gradient of this loss with respect to the logits z yields

$$\nabla_{Z_r} \mathcal{L}(Z_r, y) = P_{1|r} - e_y \,,$$

and the per-sample gradient is given by

$$g_{r,y} \;=\; \nabla_\theta \mathcal{L}(Z_r, y) \;=\; (\nabla_\theta Z_r)^\top (P_{1|r} - e_y), \tag{21}$$

where e_y is the one-hot vector for token y.

Base Covariance and Gradient Jacobians. When $\varepsilon = 0$, routing is deterministic ($Z_r = Z_0$), yielding the wake-phase base gradient covariance:

$$\Sigma_0 = \mathrm{Cov}_{y \sim P_{T|0}}(g_{0,y}) = (\nabla_\theta Z_0)^\top \tilde{H}_T (\nabla_\theta Z_0) \succeq 0, \tag{22}$$

where $\tilde{H}_T = \mathrm{diag}(P_{T|0}) - P_{T|0} P_{T|0}^\top$ is the temperature-T softmax Hessian.

Let $J_{\mathrm{upper}} = \nabla_{h_\ell} \Phi_{\mathrm{upper}}\big|_{h_\ell^{\mathrm{base}}} \in \mathbb{R}^{|\mathcal{V}| \times d_\ell}$ and $J_\theta = \nabla_\theta h_\ell^{\mathrm{base}} \in \mathbb{R}^{d_\ell \times p}$. Define the expert covariance $\Sigma_h := \mathrm{Cov}_r(\Delta h_r) \in \mathbb{R}^{d_\ell \times d_\ell}$, the temperature-1 softmax Hessian $H_0 := \mathrm{diag}(P_{1|0}) - P_{1|0} P_{1|0}^\top$, and the matrices:

$$\mathcal{M} = J_\theta^\top J_{\mathrm{upper}}^\top H_0 J_{\mathrm{upper}}, \qquad \tilde{\mathcal{M}}_T = \frac{1}{T} J_\theta^\top J_{\mathrm{upper}}^\top \tilde{H}_T J_{\mathrm{upper}}. \tag{23}$$

Theorem 6 (Random Expert Injection Improves Gradient Covariance Conditioning)**.** *Under the Gauss–Newton approximation ($\nabla_\theta Z_r \approx \nabla_\theta Z_0$) and the zero-mean expert displacement assumption ($\mathbb{E}_r[\Delta h_r] = 0$; see Remark 8), the gradient covariance under random expert injection, $\Sigma_{\mathrm{raw}} := \mathrm{Cov}(g_{r,y})$ decomposes as:*

$$\Sigma_{\mathrm{raw}} = \Sigma_0 + \varepsilon^2 \, \Sigma_{\mathrm{inject}} + R(\varepsilon), \tag{24}$$

where $\|R(\varepsilon)\| \le C\varepsilon^3$ for a constant $C > 0$, and Σ_{inject} decomposes into between-expert and within-expert terms:

$$\Sigma_{\mathrm{inject}} = \underbrace{(\mathcal{M} - \tilde{\mathcal{M}}_T)\, \Sigma_h \, (\mathcal{M} - \tilde{\mathcal{M}}_T)^\top}_{\Sigma_{\mathrm{between}} \succeq 0} + \Sigma_{\mathrm{within}}, \tag{25}$$

with $\|\Sigma_{\mathrm{within}}\| = O(\|\Sigma_h\|)$. Consequently:

- *(i)* ***Löwner Ordering:*** *For any unit vector $w \in \mathbb{R}^p$, $\Sigma_{\mathrm{raw}} \succeq \Sigma_0 + \varepsilon^2(\Sigma_{\mathrm{between}} - \|\Sigma_{\mathrm{within}}\| I) - C\varepsilon^3 I$. Along any direction $v \in \mathrm{range}(\Sigma_{\mathrm{between}})$ where $\lambda_{\min}^+(\Sigma_{\mathrm{between}}) > \|\Sigma_{\mathrm{within}}\|$, the variance is strictly boosted.*
- *(ii)* ***Subspace Confinement and Eigenvalue Amplification:*** *For every $w \in \mathrm{null}(\Sigma_0)$, $w^\top \Sigma_{\mathrm{inject}} w = 0$. Expert injection is strictly confined to $\mathcal{S} = \mathrm{range}(\Sigma_0)$ and cannot expand its rank; instead, it amplifies eigenvalues within $\mathcal{S}$.*
- *(iii)* ***Conditioning Improvement on Support:*** *Restricted to $\mathcal{S} = \mathrm{range}(\Sigma_0)$, if $\Sigma_{\mathrm{inject}}|_{\mathcal{S}} \succeq 0$ and $\kappa(\Sigma_{\mathrm{inject}}|_{\mathcal{S}}) < \kappa(\Sigma_0|_{\mathcal{S}})$, then for all sufficiently small $\varepsilon > 0$, $\kappa(\Sigma_{\mathrm{raw}}|_{\mathcal{S}}) < \kappa(\Sigma_0|_{\mathcal{S}})$.*

Gradient-Based Importance Filtering. Generated dreams are filtered by retaining points satisfying $\|g_d\|^2 \ge \tau^2$. Under the Gaussian gradient model $g \sim \mathcal{N}(0, \Sigma_{\mathrm{raw}})$ (see Remark 11), we evaluate $\Sigma_{\mathrm{filtered}} = \mathbb{E}[gg^\top \mid \|g\|^2 \ge \tau^2]$.

Theorem 7 (Spectral Preservation and Amplification)**.** *Let $g \sim \mathcal{N}(0, \Sigma_{\mathrm{raw}})$ with eigendecomposition $\Sigma_{\mathrm{raw}} = U \Lambda U^\top$, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$ with $\lambda_1 \ge \cdots \ge \lambda_p > 0$. Then:*

- *(i) $\Sigma_{\mathrm{filtered}} = U \tilde{\Lambda} U^\top$ (exact eigenbasis preservation).*
- *(ii) $\tilde{\lambda}_i > \lambda_i$ for all $i \in \{1, \ldots, p\}$ (strict eigenvalue amplification).*
- *(iii) The amplification ratio $r_i := \tilde{\lambda}_i / \lambda_i$ is monotonically non-decreasing in λ_i: $\lambda_i \ge \lambda_j \implies r_i \ge r_j$.*

Part (iii) shows that gradient filtering is not merely a variance inflator: it *preferentially* amplifies eigenvalues that are already large. This might seem counterproductive for conditioning, but it is precisely the right behavior when combined with the ReSTEM reward step, which rectifies the gradient distribution along the task-relevant descent direction. The net effect is that the filtering step amplifies the signal (high-eigenvalue directions carry task-relevant information) while the reward step removes the noise.

SEAL ReSTEM Reward as Geometric Projection. The SEAL framework (Zweiger et al., 2025) performs self-improvement via a reward-weighted inner loop. For each synthetic datum d generated by the dreaming process:

1. **Inner-loop adaptation**: $\theta_{\text{adapted}} = \theta - \eta\, g_d$, where $g_d = \nabla_\theta \mathcal{L}(d;\theta)$.
2. **Binary reward**: evaluate the adapted model on a held-out context C:

$$r(d) \;=\; \mathbf{1}\big[\mathcal{L}_{\text{task}}(\theta_{\text{adapted}};\, C) < \mathcal{L}_{\text{task}}(\theta;\, C)\big]. \tag{26}$$

3. **First-order approximation**: Taylor-expanding around θ,

$$\mathcal{L}_{\text{task}}(\theta - \eta g_d;\, C) \approx \mathcal{L}_{\text{task}}(\theta;\, C) - \eta\, g_C^\top g_d, \tag{27}$$

where $g_C = \nabla_\theta \mathcal{L}_{\text{task}}(\theta;\, C)$. Hence

$$r(d) = 1 \quad \Longleftrightarrow \quad g_C^\top g_d > 0. \tag{28}$$

The expected update direction, averaging over accepted data, is

$$\bar{g}_{\text{update}} \;=\; \mathbb{E}\big[g_d \mid g_C^\top g_d > 0\big]. \tag{29}$$

Proposition 1 (Half-Space Expectation)**.** *If $g_d \sim \mathcal{N}(0, \Sigma_{\text{filtered}})$, the expected update direction over accepted dreams is:*

$$\bar{g}_{\text{update}} = \mathbb{E}[g_d \mid g_C^\top g_d > 0] = \sqrt{\frac{2}{\pi}}\, \frac{\Sigma_{\text{filtered}}\, g_C}{\sqrt{g_C^\top \Sigma_{\text{filtered}}\, g_C}}. \tag{30}$$

The expected task loss reduction is $\mathbb{E}[\Delta\mathcal{L}_{\text{task}}] \approx -\eta\sqrt{2/\pi}\sqrt{g_C^\top \Sigma_{\text{filtered}} g_C}$.

The loss reduction is proportional to $\sqrt{g_C^\top \Sigma_{\text{filtered}}\, g_C}$, which is the Mahalanobis norm of the task gradient under the filtered covariance. A well-conditioned Σ_{filtered} ensures that the loss reduction does not collapse even when g_C is not aligned with the top eigenvector.

Theorem 8 (Spectral Whitening via Injection and Reward)**.** *Let $\mathcal{S} = \text{range}(\Sigma_0)$ with $s = \dim(\mathcal{S})$. Write $\mu_1 = \lambda_{\max}(\Sigma_0|_{\mathcal{S}})$ and $\mu_s = \lambda_{\min}(\Sigma_0|_{\mathcal{S}}) > 0$. Let $\delta_1 = \lambda_{\max}(\Sigma_{\text{inject}}|_{\mathcal{S}})$ and $\delta_s = \lambda_{\min}(\Sigma_{\text{inject}}|_{\mathcal{S}})$. If the task gradient g_C aligns with the dominant eigenvector of $\Sigma_{\text{raw}}|_{\mathcal{S}}$, then:*

$$\kappa^+(\Sigma_{\text{dream}}) \le \max\!\left(1 - \frac{2}{\pi},\, \frac{\lambda_2}{\lambda_1}\right) \frac{\mu_1 + \varepsilon^2\delta_1}{\mu_s + \varepsilon^2\delta_s}, \tag{31}$$

where $\lambda_1 \ge \lambda_2$ are the top two eigenvalues of $\Sigma_{\text{raw}}|_{\mathcal{S}}$. In particular, when $\lambda_2/\lambda_1 \le 1 - 2/\pi$ and $\kappa^+(\Sigma_{\text{inject}}|_{\mathcal{S}}) < \kappa^+(\Sigma_0)$ (i.e., $\delta_s/\delta_1 > \mu_s/\mu_1$):

$$\kappa^+(\Sigma_{\text{dream}}) \le \left(1 - \frac{2}{\pi}\right) \frac{\mu_1 + \varepsilon^2\delta_1}{\mu_s + \varepsilon^2\delta_s} < \left(1 - \frac{2}{\pi}\right) \frac{\mu_1}{\mu_s} \approx 0.363\, \kappa^+(\Sigma_0). \tag{32}$$

When gradient filtering is applied prior to ReSTEM, the bound becomes:

$$\kappa^+(\Sigma_{\text{dream}}) \le \max\!\left(1 - \frac{2}{\pi},\, \frac{\lambda_2}{\lambda_1}\right) \frac{r_1}{r_p}\, \frac{\mu_1 + \varepsilon^2\delta_1}{\mu_s + \varepsilon^2\delta_s}. \tag{33}$$

where $r_1 \ge r_p > 1$ are the amplification ratios from Theorem 7.

Discussion and Analytical Remarks.

Remark 7 (Role of the Gauss–Newton Approximation)**.** *Treating $\nabla_\theta Z_r \approx \nabla_\theta Z_0$ – is the step that makes $\mathcal{M}$ independent of y, which in turn makes the sampling noise and the expert-injection noise independent. Without this approximation, the gradient perturbation $\varepsilon\, \mathcal{M}(y)\, \Delta h_r$ depends on y through a Hessian-correction term $\sum_c (P^c_{1|0} - e^c_y)\, \nabla_{h_\ell}[(\nabla_\theta Z)_{c,:}]$, which couples the two noise sources and prevents a clean additive decomposition. The Gauss–Newton approximation is standard in the deep learning theory literature and is most accurate when the model is well-trained (residuals $P_{1|0} - e_y$ are small) or when $T \approx 1$.*

Remark 8 (Zero-Mean Expert Displacement)**.** *The condition $\mathbb{E}_r[\Delta h_r] = 0$ holds whenever non-selected experts are unbiased relative to selected experts. In practice, with large expert counts $E \gg K$, both the gating-weighted average and uniform average over remaining experts converge to the population mean.*

Remark 9 (Subspace Confinement Mechanism)**.** *Because all covariance terms pass through the logit Jacobian $J_{\text{upper}} J_\theta$ and the softmax Hessian null space is $\text{span}(\mathbf{1})$, perturbations that shift all logits uniformly have zero gradient. Hence, expert injection cannot activate directions outside the gradient support $\mathcal{S}$; its role is eigenvalue redistribution within $\mathcal{S}$.*

Remark 10 (Why Random Injection Improves Conditioning in Practice)**.** *The base covariance Σ_0 inherits severe ill-conditioning from the loss landscape. In contrast, the expert covariance Σ_h across independently specialized experts is near-isotropic on its support. Projecting Σ_h into parameter space boosts trailing eigenvalues, inflating variance along poorly conditioned directions.*

Remark 11 (Justification of the Gaussian Gradient Model)**.** *Since the token error $(P_{1|0} - e_y)$ is bounded in $[-1, 1]$, gradients are sub-Gaussian with finite moments. Moreover, mini-batch gradient averaging across tokens invokes the multivariate Central Limit Theorem, making the Gaussian approximation accurate for batch training.*

Remark 12 (Distributional Robustness of Filtering Analysis)**.** *In Theorem 7, Part (i) (eigenbasis preservation) requires only coordinate independence and symmetry under reflections; Part (ii) (strict amplification) requires only support beyond τ; only Part (iii) (monotonicity of ratios) relies specifically on Gaussianity.*

Remark 13 (Complementary Roles in the Dreaming Pipeline)**.** *The dreaming pipeline combines three complementary mechanisms:*

(i) ***Random MoE Injection*** *inflates trailing eigenvalues, improving baseline conditioning: $\kappa^+(\Sigma_{\text{raw}}) < \mu_1/\mu_s$.*

(ii) ***ReSTEM Reward*** *applies a rank-1 downdate that selectively reduces the dominant eigenvalue by $(1 - 2/\pi) \approx 0.36$.*

(iii) ***Gradient Filtering*** *has a three-fold impact:*

- ***Sample efficiency.*** *Most dream-generated data points have small gradient norms ("easy" examples the model has already learned). Discarding them focuses computation on informative examples, reducing the number of ReSTEM evaluations needed.*
- ***Robustness beyond first order.*** *The ReSTEM reward equation 26 is a binary comparison of the adapted-model loss against the base-model loss. While the first-order approximation equation 28 makes this a pure sign condition $g_C^\top g_d > 0$ (independent of $\|g_d\|$), in practice the loss difference $\mathcal{L}(\theta - \eta g_d) - \mathcal{L}(\theta)$ is $O(\eta\|g_d\|)$. When $\|g_d\|$ is very small, this difference falls below evaluation noise, making the binary reward unreliable. Filtering ensures that accepted data points produce sufficiently large adaptation steps for meaningful reward signals.*
- ***Convergence speed.*** *As discussed after Proposition 1, the expected task loss reduction per accepted datum scales as $\sqrt{g_C^\top \Sigma_{\text{filtered}}\, g_C}$. Since filtering amplifies all eigenvalues of the covariance ($\tilde{\lambda}_i > \lambda_i$), it increases the expected step size, accelerating convergence even though it degrades the condition number.*

A.3 Auxiliary Results: KL Divergence and Logit Distance

In this section, we provide formal proofs relating token-level KL divergence to ℓ_2 logit distances, rigorously justifying the quadratic logit approximation in Knowledge Seeding.

Throughout this section, let $Z_T, Z_S \in \mathbb{R}^{|\mathcal{V}|}$ be teacher and student logits, $P_T = \text{softmax}(Z_T)$, $P_S = \text{softmax}(Z_S)$, $\delta = Z_S - Z_T$, and $\hat{\delta} = \delta - \bar{\delta}\mathbf{1}$ with $\bar{\delta} = \frac{1}{|\mathcal{V}|}\sum_v \delta_v$.

Lemma 9 (KL Divergence as Bregman Divergence of Log-Sum-Exp)**.** *Let $A(Z) = \log \sum_{v \in \mathcal{V}} \exp(Z_v)$ be the log-partition function. Then:*

$$D_{\text{KL}}(P_T \| P_S) = A(Z_S) - A(Z_T) - \nabla A(Z_T)^\top (Z_S - Z_T) = \int_0^1 (1-t)\, \delta^\top \Sigma(P_{(t)})\, \delta \, dt, \quad (34)$$

where $P_{(t)} = \mathrm{softmax}(Z_T + t\delta)$ *and* $\Sigma(P) = \mathrm{diag}(P) - PP^\top$ *is the softmax Hessian.*

Proof. By direct expansion of the definition of KL divergence:

$$D_{\mathrm{KL}}(P_T \| P_S) = \sum_{v\in\mathcal{V}} P_{T,v} \log \frac{P_{T,v}}{P_{S,v}} = \sum_{v\in\mathcal{V}} P_{T,v}\big[(Z_T)_v - A(Z_T) - (Z_S)_v + A(Z_S)\big]$$
$$= P_T^\top (Z_T - Z_S) + A(Z_S) - A(Z_T) = A(Z_S) - A(Z_T) - \nabla A(Z_T)^\top \delta,$$

using $\nabla A(Z_T) = P_T$. Applying Taylor's theorem with integral remainder to $A(Z)$ along $Z(t) = Z_T + t\delta$ yields the integral form since $\nabla^2 A(Z(t)) = \Sigma(P_{(t)})$. □

Lemma 10 (Spectral Properties of the Softmax Hessian)**.** *For any probability vector* $P \in \Delta^{|\mathcal{V}|-1}$ *with* $p_{\min} = \min_v P_v > 0$*, the softmax Hessian* $\Sigma(P) = \mathrm{diag}(P) - PP^\top$ *satisfies:*

(a) $\Sigma(P) \preceq \frac{1}{2} I$ *(all eigenvalues are at most* $1/2$*).*

(b) $\Sigma(P)\mathbf{1} = 0$ *(*$\mathbf{1}$ *spans the null space).*

(c) $\lambda^+_{\min}(\Sigma(P)) \geq p_{\min}$ *(the smallest non-zero eigenvalue is at least* $p_{\min}$*).*

Proof. **Part (a):** For any unit vector $u \in \mathbb{R}^{|\mathcal{V}|}$ with $\|u\| = 1$:

$$u^\top \Sigma(P) u = \sum_v P_v u_v^2 - \Big(\sum_v P_v u_v\Big)^2 = \mathrm{Var}_{v\sim P}(u_v).$$

Let $u_{\max} = \max_v u_v$ and $u_{\min} = \min_v u_v$. By Popoviciu's inequality on the variance of a bounded random variable, $\mathrm{Var}_{v\sim P}(u_v) \leq \frac{1}{4}(u_{\max} - u_{\min})^2$. By the AM-QM inequality, $(u_{\max} - u_{\min})^2 \leq 2(u_{\max}^2 + u_{\min}^2) \leq 2\|u\|^2 = 2$. Hence $u^\top \Sigma(P) u \leq \frac{1}{2}$, establishing $\Sigma(P) \preceq \frac{1}{2} I$.

Part (b): $\Sigma(P)\mathbf{1} = \mathrm{diag}(P)\mathbf{1} - P(P^\top \mathbf{1}) = P - P(1) = 0$.

Part (c): Let $u \perp \mathbf{1}$ with $\|u\| = 1$, and let $c = \sum_v P_v u_v$. Then:

$$u^\top \Sigma(P) u = \sum_v P_v (u_v - c)^2 \geq p_{\min} \sum_v (u_v - c)^2 = p_{\min} \|u - c\mathbf{1}\|^2.$$

Since $u \perp \mathbf{1}$, $\|u - c\mathbf{1}\|^2 = \|u\|^2 - 2c\langle u, \mathbf{1}\rangle + c^2|\mathcal{V}| = 1 + c^2|\mathcal{V}| \geq 1$. Thus $u^\top \Sigma(P) u \geq p_{\min}$ for all unit vectors $u \perp \mathbf{1}$, proving $\lambda^+_{\min}(\Sigma(P)) \geq p_{\min}$. □

Theorem 11 (Upper Bound on KL Divergence via Logit Distance)**.** *For any teacher and student logits* $Z_T, Z_S \in \mathbb{R}^{|\mathcal{V}|}$*:*

$$D_{\mathrm{KL}}(P_T \| P_S) \leq \frac{1}{4}\|\hat{\delta}\|^2 \leq \frac{1}{4}\|\delta\|^2, \tag{35}$$

where $\hat{\delta} = \delta - \bar{\delta}\mathbf{1}$ *is the centered logit difference.*

Proof. From the integral representation in Lemma 9:

$$D_{\mathrm{KL}}(P_T \| P_S) = \int_0^1 (1-t)\, \delta^\top \Sigma(P_{(t)})\, \delta \, dt.$$

Since $\Sigma(P_{(t)})\mathbf{1} = 0$ by Lemma 10(b), $\delta^\top \Sigma(P_{(t)})\delta = \hat{\delta}^\top \Sigma(P_{(t)})\hat{\delta}$. Applying $\Sigma(P_{(t)}) \preceq \frac{1}{2} I$ from Lemma 10(a):

$$D_{\mathrm{KL}}(P_T \| P_S) \leq \int_0^1 (1-t)\, \frac{1}{2}\|\hat{\delta}\|^2 \, dt = \frac{1}{4}\|\hat{\delta}\|^2.$$

Finally, $\|\hat{\delta}\|^2 = \|\delta - \bar{\delta}\mathbf{1}\|^2 = \|\delta\|^2 - |\mathcal{V}|\bar{\delta}^2 \leq \|\delta\|^2$. □

Remark 14 (Justification for Squared-Norm Loss)**.** *Theorem 11 proves that minimizing the squared logit distance provides a rigorous, conservative upper bound on the true KL divergence.*

A.4 Proofs of Theorems in Memory Consolidation

A.4.1 Proof of Theorem 4 (Quadratic Form of the KS Loss)

Proof. To lighten the notation, we let $J = J_{h^d_{\ell-1}}$, $J' = J_{h^s_{\ell-1}}$, $h = h^d_{\ell-1}$, $h' = h^s_{\ell-1}$, and $u = u_{h^d_{\ell-1}}$, $u' = u_{h^s_{\ell-1}}$. The loss is given by

$$\mathcal{L}(M) := \alpha(1-\lambda)\|JMh - u\|^2 + \mathbb{E}_{h'}\|J'Mh' - u'\|^2_Q$$

Expanding the first term we have

$$\begin{aligned}\|JMh - u\|^2 &= h^\top M^\top J^\top JMh - 2u^\top JMh + \|u\|^2 \\ &= \mathrm{Tr}(M^\top J^\top JMhh^\top) - 2\,\mathrm{Tr}(hu^\top JM) + \|u\|^2\end{aligned}$$

using the cyclic property of trace operator. We next use the identity $\mathrm{Tr}(X^\top AXB) = \mathrm{vec}(X)^\top(B^\top \otimes A)\mathrm{vec}(X)$ and $\mathrm{Tr}(A^\top B) = \mathrm{vec}(A)^\top \mathrm{vec}(B)$, and write

$$\|JMh - u\|^2 = \mathrm{vec}(M)^\top((hh^\top)\otimes(J^\top J))\mathrm{vec}(M) - 2\mathrm{vec}(J^\top uh^\top)^\top \mathrm{vec}(M) + \|u\|^2$$

Likewise the second term in the loss can be written as

$$\begin{aligned}\mathbb{E}_{h'}\|J'Mh' - u'\|^2_Q = \mathrm{vec}(M)^\top \mathbb{E}_{h'}\left[(h'h'^\top)\otimes(J'^\top QJ')\right]\mathrm{vec}(M) \\ - 2\mathbb{E}_{h'}\left[\mathrm{vec}\big(J'^\top Qu'h'^\top\big)\right]^\top \mathrm{vec}(M) + \mathbb{E}_{h'}\|u'\|^2_Q\end{aligned}$$

Combining the two terms, we get

$$\mathcal{L}(M) = m^\top Hm - 2c^\top m + \alpha(1-\lambda)\|u\|^2 + \mathbb{E}_{h'}\|u'\|^2_Q$$

where $m := \mathrm{vec}(M)$ and

$$\begin{aligned}H &= \alpha(1-\lambda)((hh^\top)\otimes(J^\top J)) + \mathbb{E}_{h'}\left[(h'h'^\top)\otimes(J'^\top QJ')\right] \\ c &= \alpha(1-\lambda)\mathrm{vec}(J^\top uh^\top) + \mathbb{E}_{h'}\left[\mathrm{vec}\big(J'^\top Qu'h'^\top\big)\right]\end{aligned}$$

This completes the proof. □

A.4.2 Proof of Theorem 5 (SVD Closed-Form Solution)

Proof. Under a linear upper network W_u and $\lambda = 1$, the loss is:

$$\begin{aligned}\mathcal{L}_{\mathrm{KS}}(M) &= \mathbb{E}_{h^s_{\ell-1}}\left[(W_uMh^s_{\ell-1} - u_{h^s_{\ell-1}})^\top Q(W_uMh^s_{\ell-1} - u_{h^s_{\ell-1}})\right] \\ &= \mathrm{Tr}\left(M^\top W_u^\top QW_uM\,\mathbb{E}_{h^s_{\ell-1}}[h^s_{\ell-1}(h^s_{\ell-1})^\top]\right) - 2\,\mathrm{Tr}\left(M^\top W_u^\top Q\,\mathbb{E}_{h^s_{\ell-1}}[u_{h^s_{\ell-1}}(h^s_{\ell-1})^\top]\right) + \mathrm{const.}\end{aligned}$$

Recalling $H_S = \mathbb{E}_{h^s_{\ell-1}}[h^s_{\ell-1}(h^s_{\ell-1})^\top]$, $\tilde{Q} = W_u^\top QW_u$, and $C = W_u^\top Q\,\mathbb{E}_{h^s_{\ell-1}}[u_{h^s_{\ell-1}}(h^s_{\ell-1})^\top]$:

$$\mathcal{L}_{\mathrm{KS}}(M) = \mathrm{Tr}\left(M^\top \tilde{Q}MH_S\right) - 2\,\mathrm{Tr}\left(M^\top C\right) + \mathrm{const.}$$

Because $Q \succ 0$ and W_u has full column rank, $\tilde{Q} \succ 0$ is symmetric positive definite. Since $H_S \succ 0$, we can complete the square in Frobenius norm:

$$\mathcal{L}_{\mathrm{KS}}(M) = \left\|\tilde{Q}^{1/2}MH_S^{1/2} - \tilde{Q}^{-1/2}CH_S^{-1/2}\right\|^2_F + \mathrm{const}'.$$

Finding the optimal rank-r matrix M is equivalent to minimizing $\|\tilde{Q}^{1/2}MH_S^{1/2} - X\|^2_F$ subject to $\mathrm{rank}(M) \le r$, where $X = \tilde{Q}^{-1/2}CH_S^{-1/2}$. By the Eckart-Young-Mirsky theorem (Eckart & Young, 1936; Mirsky, 1960), the optimal rank-r approximation of X in Frobenius norm is $\mathrm{SVD}_r(X)$. Setting $\tilde{Q}^{1/2}M_*H_S^{1/2} = \mathrm{SVD}_r(X)$ and isolating M_* yields:

$$M_* = \tilde{Q}^{-1/2}\,\mathrm{SVD}_r\left(\tilde{Q}^{-1/2}CH_S^{-1/2}\right)H_S^{-1/2},$$

which completes the proof. □

A.5 Proofs of Theorems in the Dreaming Phase

A.5.1 Proof of Theorem 6 (Random Expert Injection Decomposition and Conditioning)

Proof. **Step 1: Linearized Gradient Expansion.** Taylor expanding the logits around $\varepsilon = 0$:

$$Z_r = \Phi_{\text{upper}}(h_\ell^{\text{base}} + \varepsilon \Delta h_r) = Z_0 + \varepsilon J_{\text{upper}} \Delta h_r + O(\varepsilon^2).$$

Expanding the predictive distribution $P_{1|r} = \text{softmax}(Z_r)$ around Z_0:

$$P_{1|r} = P_{1|0} + \varepsilon H_0 J_{\text{upper}} \Delta h_r + O(\varepsilon^2),$$

where $H_0 = \text{diag}(P_{1|0}) - P_{1|0} P_{1|0}^\top$. Under the Gauss–Newton approximation $\nabla_\theta Z_r \approx \nabla_\theta Z_0 = J_{\text{upper}} J_\theta$, and the per-sample gradient equation 21 expands as:

$$g_{r,y} = \underbrace{J_\theta^\top J_{\text{upper}}^\top (P_{1|0} - e_y)}_{g_{0,y} \text{ (depends on } y \text{ only)}} + \varepsilon \underbrace{J_\theta^\top J_{\text{upper}}^\top H_0 J_{\text{upper}}}_{\mathcal{M} \in \mathbb{R}^{p \times d_\ell} \text{ (constant)}} \underbrace{\Delta h_r}_{\text{depends on } r \text{ only}} + O(\varepsilon^2). \tag{36}$$

Step 2: Law of Total Covariance. By the law of total covariance:

$$\text{Cov}(g_{r,y}) = \text{Cov}_r\big(\mathbb{E}[g_{r,y} \mid r]\big) + \mathbb{E}_r\big[\text{Cov}(g_{r,y} \mid r)\big].$$

Between-expert covariance: Conditioned on expert r, the mean gradient is $\mathbb{E}[g_{r,y} \mid r] = (\nabla_\theta Z_0)^\top (P_{1|r} - P_{T|r})$. Expanding $P_{T|r} = \text{softmax}(Z_r/T) = P_{T|0} + \frac{\varepsilon}{T} \tilde{H}_T J_{\text{upper}} \Delta h_r + O(\varepsilon^2)$ where $\tilde{H}_T = \text{diag}(P_{T|0}) - P_{T|0} P_{T|0}^\top$, we have:

$$\mathbb{E}[g_{r,y} \mid r] = \mu_0 + \varepsilon(\mathcal{M} - \tilde{\mathcal{M}}_T) \Delta h_r + O(\varepsilon^2),$$

where $\mu_0 = (\nabla_\theta Z_0)^\top (P_{1|0} - P_{T|0})$ and $\tilde{\mathcal{M}}_T = \frac{1}{T} J_\theta^\top J_{\text{upper}}^\top \tilde{H}_T J_{\text{upper}}$. Taking the covariance over r with $\mathbb{E}_r[\Delta h_r] = 0$:

$$\text{Cov}_r\big(\mathbb{E}[g_{r,y} \mid r]\big) = \varepsilon^2 (\mathcal{M} - \tilde{\mathcal{M}}_T) \Sigma_h (\mathcal{M} - \tilde{\mathcal{M}}_T)^\top + O(\varepsilon^3) =: \varepsilon^2 \Sigma_{\text{between}} + O(\varepsilon^3).$$

Since $\Sigma_h = \text{Cov}_r(\Delta h_r) \succeq 0$, the matrix $\Sigma_{\text{between}} = (\mathcal{M} - \tilde{\mathcal{M}}_T) \Sigma_h (\mathcal{M} - \tilde{\mathcal{M}}_T)^\top$ is positive semi-definite ($\Sigma_{\text{between}} \succeq 0$).

Within-expert covariance: Conditioned on r, the only randomness is $y \sim P_{T|r}$. The covariance of the one-hot draw e_y is $\text{Cov}_{y \sim P_{T|r}}(e_y) = \text{diag}(P_{T|r}) - P_{T|r} P_{T|r}^\top =: \tilde{H}_T^{(r)}$. Thus:

$$\text{Cov}(g_{r,y} \mid r) = (\nabla_\theta Z_0)^\top \tilde{H}_T^{(r)} (\nabla_\theta Z_0).$$

Softmax Taylor expansion. We now expand $P_{T|r} = \text{softmax}(Z_0/T + \varepsilon\, \delta/T)$ to second order in ε, where $\delta = J_{\text{upper}}\, \Delta h_r$. For the v-th component, write $f(\varepsilon) = P_{T|r,v}(\varepsilon)$ via the log-partition function $L(\varepsilon) = \log \sum_k e^{z_k + \varepsilon \delta_k / T}$ (with $z = Z_0/T$):

$$\log f(\varepsilon) = z_v + \frac{\varepsilon\, \delta_v}{T} - L(\varepsilon), \qquad f(\varepsilon) = p_v\, e^{\varepsilon \delta_v / T - [L(\varepsilon) - L(0)]},$$

where $p_v = P_{T|0,v} = f(0)$. The derivatives of the log-partition function are standard cumulant-generating identities:

$$L'(\varepsilon) = \frac{1}{T} \sum_k p_k(\varepsilon)\, \delta_k = \frac{\bar{\delta}(\varepsilon)}{T}, \qquad L''(\varepsilon) = \frac{1}{T^2} \sum_k p_k(\varepsilon)\, (\delta_k - \bar{\delta}(\varepsilon))^2 = \frac{\sigma_\delta^2(\varepsilon)}{T^2},$$

where $p_k(\varepsilon) = P_{T|r,k}(\varepsilon)$. At $\varepsilon = 0$: $L'(0) = \bar{\delta}/T$ and $L''(0) = \sigma_\delta^2/T^2$, with $\bar{\delta} = \sum_k p_k \delta_k$ and $\sigma_\delta^2 = \sum_k p_k (\delta_k - \bar{\delta})^2$.

First order. Differentiating $\log f(\varepsilon) = z_v + \varepsilon \delta_v / T - L(\varepsilon)$ yields $f'(\varepsilon)/f(\varepsilon) = \delta_v/T - L'(\varepsilon)$, i.e.,

$$f'(\varepsilon) = f(\varepsilon) \cdot \big(\delta_v/T - L'(\varepsilon)\big). \tag{37}$$

Evaluating at $\varepsilon = 0$:

$$f'(0) = p_v \cdot \left(\frac{\delta_v}{T} - \frac{\bar{\delta}}{T} \right) = \frac{p_v}{T} (\delta_v - \bar{\delta}) =: \eta_{r,v}.$$

In vector form, $\eta_r = \frac{1}{T} \tilde{H}_T \, \delta$, since $[\tilde{H}_T \, \delta]_v = [\mathrm{diag}(P_{T|0}) - P_{T|0} P_{T|0}^\top] \delta \big|_v = p_v \delta_v - p_v \bar{\delta} = p_v (\delta_v - \bar{\delta})$.

Second order. Differentiating equation 37 by the product rule:

$$f''(\varepsilon) = f'(\varepsilon) \cdot \big(\delta_v / T - L'(\varepsilon) \big) \; + \; f(\varepsilon) \cdot (-L''(\varepsilon)).$$

Substituting $f'(0) = \frac{p_v}{T} (\delta_v - \bar{\delta})$, $f(0) = p_v$, $L'(0) = \bar{\delta}/T$, and $L''(0) = \sigma_\delta^2 / T^2$:

$$f''(0) = \frac{p_v}{T} (\delta_v - \bar{\delta}) \cdot \frac{(\delta_v - \bar{\delta})}{T} \; - \; p_v \cdot \frac{\sigma_\delta^2}{T^2} = \frac{p_v}{T^2} \big[(\delta_v - \bar{\delta})^2 - \sigma_\delta^2 \big].$$

So the second-order coefficient is $\xi_{r,v} = \frac{1}{2} f''(0) = \frac{p_v}{2T^2} [(\delta_v - \bar{\delta})^2 - \sigma_\delta^2]$. As a sanity check, $\sum_v \xi_{r,v} = \frac{1}{2T^2} \sum_v p_v [(\delta_v - \bar{\delta})^2 - \sigma_\delta^2] = \frac{1}{2T^2} [\sigma_\delta^2 - \sigma_\delta^2] = 0$, confirming that probabilities are conserved to second order.

Expanding $\tilde{H}_T^{(r)}$. With $P_{T|r} = P_{T|0} + \varepsilon \, \eta_r + \varepsilon^2 \, \xi_r + O(\varepsilon^3)$, we use the linearity of $\mathrm{diag}(\cdot)$ and the bilinearity of the outer product to expand $\tilde{H}_T^{(r)} = \mathrm{diag}(P_{T|r}) - P_{T|r} P_{T|r}^\top$:

$$\tilde{H}_T^{(r)} = \tilde{H}_T + \varepsilon \, F_r + \varepsilon^2 \, G_r + O(\varepsilon^3), \tag{38}$$

where

$$F_r = \mathrm{diag}(\eta_r) - P_{T|0} \eta_r^\top - \eta_r P_{T|0}^\top \qquad \text{(linear in } \Delta h_r\text{)}, \tag{39}$$

$$G_r = \mathrm{diag}(\xi_r) - P_{T|0} \xi_r^\top - \xi_r P_{T|0}^\top - \eta_r \eta_r^\top \qquad \text{(quadratic in } \Delta h_r\text{)}. \tag{40}$$

The $-\eta_r \eta_r^\top$ in equation 40 arises from the cross-term in $(P_{T|0} + \varepsilon \eta_r + \cdots)(P_{T|0} + \varepsilon \eta_r + \cdots)^\top$.

Averaging over experts. Since F_r is linear in Δh_r and $\mathbb{E}_r[\Delta h_r] = 0$, we have $\mathbb{E}_r[F_r] = 0$. However, G_r is quadratic in Δh_r, so $\mathbb{E}_r[G_r] \neq 0$. The dominant contribution is

$$\mathbb{E}_r[\eta_r \eta_r^\top] = \frac{1}{T^2} \tilde{H}_T J_{\text{upper}} \, \mathbb{E}_r[\Delta h_r \, \Delta h_r^\top] \, J_{\text{upper}}^\top \tilde{H}_T = \frac{1}{T^2} \tilde{H}_T J_{\text{upper}} \Sigma_h \, J_{\text{upper}}^\top \tilde{H}_T \; \neq \; 0.$$

Therefore

$$\mathbb{E}_r[\mathrm{Cov}(g_{r,y} \mid r)] = \Sigma_0 + \varepsilon^2 \, \Sigma_{\text{within}} + O(\varepsilon^3), \tag{41}$$

where $\Sigma_0 = (\nabla_\theta Z_0)^\top \tilde{H}_T (\nabla_\theta Z_0) \succeq 0$ and $\Sigma_{\text{within}} = (\nabla_\theta Z_0)^\top \mathbb{E}_r[G_r] (\nabla_\theta Z_0)$. The matrix Σ_{within} contains the negative term $-\tilde{\mathcal{M}}_T \Sigma_h \tilde{\mathcal{M}}_T^\top$ alongside positive contributions from ξ_r, making it generically indefinite with bounded norm $\|\Sigma_{\text{within}}\| = O(\|\Sigma_h\|)$.

Total covariance. Summing within- and between-expert contributions gives $\Sigma_{\text{raw}} = \Sigma_0 + \varepsilon^2 (\Sigma_{\text{between}} + \Sigma_{\text{within}}) + R(\varepsilon)$, establishing equation 24.

Step 3: Subspace Confinement and Conditioning.

Part (i): Follows from $w^\top \Sigma_{\text{within}} w \geq -\|\Sigma_{\text{within}}\|$ for unit w.

Part (ii): For $w \in \mathrm{null}(\Sigma_0)$, $\tilde{H}_T^{1/2} J_{\text{upper}} J_\theta w = 0$, which implies $J_{\text{upper}} J_\theta w \in \mathrm{null}(\tilde{H}_T) = \mathrm{span}(\mathbf{1})$. Since $\mathbf{1}^\top H_0 = 0$ and $\mathbf{1}^\top \tilde{H}_T = 0$, $w^\top \mathcal{M} = 0$ and $w^\top \tilde{\mathcal{M}}_T = 0$. Thus $w^\top \Sigma_{\text{between}} w = 0$ and $w^\top \Sigma_{\text{within}} w = 0$, proving subspace confinement.

Part (iii): Let $\mathcal{S} = \mathrm{range}(\Sigma_0)$ and restrict all matrices to $\mathcal{S}$; by Part (ii), $\Sigma_{\text{inject}}|_{\mathcal{S}}$ is well-defined on this support. Assume $\Sigma_{\text{inject}}|_{\mathcal{S}} \succeq 0$. Let $\lambda_1 \geq \cdots \geq \lambda_s > 0$ and $\delta_1 \geq \cdots \geq \delta_s \geq 0$ denote the eigenvalues of $\Sigma_0|_{\mathcal{S}}$ and $\Sigma_{\text{inject}}|_{\mathcal{S}}$, where $s = \dim(\mathcal{S})$. By Weyl's inequality, the eigenvalues of $\Sigma_{\text{raw}}|_{\mathcal{S}} = \Sigma_0|_{\mathcal{S}} + \varepsilon^2 \Sigma_{\text{inject}}|_{\mathcal{S}} + R(\varepsilon)|_{\mathcal{S}}$ satisfy: $\lambda_{\max}(\Sigma_{\text{raw}}|_{\mathcal{S}}) \leq \lambda_1 + \varepsilon^2 \delta_1 + C\varepsilon^3$ and $\lambda_{\min}(\Sigma_{\text{raw}}|_{\mathcal{S}}) \geq \lambda_s + \varepsilon^2 \delta_s - C\varepsilon^3$. Thus:

$$\kappa(\Sigma_{\text{raw}}|_{\mathcal{S}}) \leq \frac{\lambda_1 + \varepsilon^2 \delta_1 + C\varepsilon^3}{\lambda_s + \varepsilon^2 \delta_s - C\varepsilon^3} = \frac{\lambda_1 + \varepsilon^2 \delta_1}{\lambda_s + \varepsilon^2 \delta_s} + O(\varepsilon^3).$$

Since $\delta_s / \delta_1 > \lambda_s / \lambda_1$, the ratio $(\lambda_1 + \varepsilon^2 \delta_1)/(\lambda_s + \varepsilon^2 \delta_s) < \lambda_1 / \lambda_s = \kappa(\Sigma_0|_{\mathcal{S}})$. Hence, for all sufficiently small $\varepsilon > 0$, $\kappa(\Sigma_{\text{raw}}|_{\mathcal{S}}) < \kappa(\Sigma_0|_{\mathcal{S}})$. □

A.5.2 Proof of Theorem 7 (Spectral Preservation, Amplification, and Monotonicity)

Proof. **Part (i):** Transform to the decorrelated basis $z = U^\top g$, so that $z \sim \mathcal{N}(0, \Lambda)$ with independent components $z_i \sim \mathcal{N}(0, \lambda_i)$. The truncation event $\|g\|^2 \geq \tau^2$ maps to $\|z\|^2 = \sum_i z_i^2 \geq \tau^2$. We must show that $\mathbb{E}[z_i z_j \mid \|z\|^2 \geq \tau^2] = 0$ for $i \neq j$.

The joint density of z is $p(z) = \prod_i p_i(z_i)$ where each p_i is the density of $\mathcal{N}(0, \lambda_i)$. The truncation domain $\mathcal{S} = \{z : \sum_i z_i^2 \geq \tau^2\}$ is symmetric under the reflection $z_j \mapsto -z_j$ (for each j). Since $p_i(-z_i) = p_i(z_i)$ and $\mathbf{1}_\mathcal{S}$ is invariant under $z_j \mapsto -z_j$, the integrand $z_i \cdot z_j \cdot p(z) \cdot \mathbf{1}_\mathcal{S}(z)$ is odd in z_j (for $i \neq j$) and hence integrates to zero. Therefore $\Sigma_{\text{filtered}} = U\,\tilde{\Lambda}\,U^\top$ with $\tilde{\Lambda}$ diagonal.

Part (ii): For the diagonal entries,

$$\tilde{\lambda}_i = \mathbb{E}\big[z_i^2 \mid \|z\|^2 \geq \tau^2\big] = \frac{\mathbb{E}\big[z_i^2\,\mathbf{1}[\|z\|^2 \geq \tau^2]\big]}{\Pr(\|z\|^2 \geq \tau^2)}. \tag{42}$$

Write $z_i^2 = z_i^2\,\mathbf{1}[\|z\|^2 \geq \tau^2] + z_i^2\,\mathbf{1}[\|z\|^2 < \tau^2]$. Taking expectations:

$$\lambda_i = \tilde{\lambda}_i\,\Pr(\|z\|^2 \geq \tau^2) + \mathbb{E}\big[z_i^2\,\mathbf{1}[\|z\|^2 < \tau^2]\big]\,. \tag{43}$$

Let $q = \Pr(\|z\|^2 \geq \tau^2) \in (0, 1)$. Then $\lambda_i = q\,\tilde{\lambda}_i + (1-q)\,\lambda_i^-$ where $\lambda_i^- = \mathbb{E}[z_i^2 \mid \|z\|^2 < \tau^2]$.

Since $\|z\|^2 < \tau^2$ restricts the magnitude of z, we have $\lambda_i^- \leq \tau^2$ and moreover $\lambda_i^- < \lambda_i$ (the conditional mean of a squared Gaussian on a bounded set is strictly less than the unconditional mean, provided $\tau^2 < \infty$). Solving: $\tilde{\lambda}_i = \lambda_i/q - (1-q)\lambda_i^-/q > \lambda_i/q - (1-q)\lambda_i/q = \lambda_i$.

Part (iii): Transform to normalized coordinates $y_i = z_i/\sqrt{\lambda_i}$, so $y_i \sim \mathcal{N}(0, 1)$ i.i.d. The truncation condition becomes $\sum_i \lambda_i y_i^2 \geq \tau^2$. The amplification ratio is

$$\frac{\tilde{\lambda}_i}{\lambda_i} = \mathbb{E}\left[y_i^2 \;\middle|\; \sum_k \lambda_k\,y_k^2 \geq \tau^2\right]. \tag{44}$$

We claim that this conditional expectation is non-decreasing in λ_i. Increasing λ_i relaxes the truncation constraint along axis i: for fixed values of $\{y_k\}_{k\neq i}$, the minimum value of $|y_i|$ needed to satisfy $\sum_k \lambda_k y_k^2 \geq \tau^2$ *decreases* as λ_i increases.

Formally, consider two indices i, j with $\lambda_i > \lambda_j$. Define the event $\mathcal{S} = \{\sum_k \lambda_k y_k^2 \geq \tau^2\}$. For any realization $(y_1, \ldots, y_p)$ satisfying the constraint, the contribution of axis i to the sum is $\lambda_i y_i^2 > \lambda_j y_i^2$ (when $y_i \neq 0$). This means the constraint is "easier" to satisfy via axis i than via axis j, so conditional on $\mathcal{S}$, the distribution of y_i^2 is stochastically larger than that of y_j^2. Hence $\mathbb{E}[y_i^2 \mid \mathcal{S}] \geq \mathbb{E}[y_j^2 \mid \mathcal{S}]$, establishing monotonicity. We next provide a rigorous proof of this insight.

Since $P(\mathcal{S}) > 0$, it suffices to show

$$N := \mathbb{E}\big[(y_i^2 - y_j^2)\mathbf{1}_\mathcal{S}(y)\big] \geq 0, \tag{45}$$

because $\mathbb{E}[y_i^2 \mid \mathcal{S}] - \mathbb{E}[y_j^2 \mid \mathcal{S}] = N/P(\mathcal{S})$.

Let $\sigma_{ij} : \mathbb{R}^p \to \mathbb{R}^p$ denote the transposition that swaps coordinates i and j:

$$[\sigma_{ij}(y)]_k = \begin{cases} y_j & \text{if } k = i, \\ y_i & \text{if } k = j, \\ y_k & \text{otherwise.} \end{cases}$$

Since $y_1, \ldots, y_p$ are i.i.d. $\mathcal{N}(0, 1)$, the joint distribution is invariant under σ_{ij}: $y \overset{d}{=} \sigma_{ij}(y)$.

Applying the substitution $y \mapsto \sigma_{ij}(y)$ in the expectation equation 45:

$$N = \mathbb{E}\big[(\sigma_{ij}(y)_i^2 - \sigma_{ij}(y)_j^2)\mathbf{1}_\mathcal{S}(\sigma_{ij}(y))\big] = \mathbb{E}\big[(y_j^2 - y_i^2)\mathbf{1}_{\mathcal{S}^\sigma}(y)\big], \tag{46}$$

where $\mathcal{S}^\sigma := \sigma_{ij}(\mathcal{S}) = \big\{\sum_{k\neq i,j} \lambda_k y_k^2 + \lambda_j y_i^2 + \lambda_i y_j^2 \geq \tau^2\big\}$ is the event $\mathcal{S}$ with the coefficients λ_i and λ_j exchanged.

We next add the original expression equation 45 and the transformed expression equation 46:

$$2N = \mathbb{E}\Big[(y_i^2 - y_j^2)\big(\mathbf{1}_{\mathcal{S}}(y) - \mathbf{1}_{\mathcal{S}^\sigma}(y)\big)\Big]. \tag{47}$$

We now show that the integrand in equation 47 is non-negative for every $y \in \mathbb{R}^p$. Decompose the quadratic sums as:

$$\sum_k \lambda_k y_k^2 = R + Q, \qquad \sum_k \lambda_k^\sigma y_k^2 = R + Q^\sigma,$$

where $R = \sum_{k \neq i,j} \lambda_k y_k^2$, $Q = \lambda_i y_i^2 + \lambda_j y_j^2$, and $Q^\sigma = \lambda_j y_i^2 + \lambda_i y_j^2$. Their difference is:

$$Q - Q^\sigma = (\lambda_i - \lambda_j)(y_i^2 - y_j^2). \tag{48}$$

Since $\lambda_i > \lambda_j$, the sign of $Q - Q^\sigma$ equals the sign of $y_i^2 - y_j^2$. Now consider all cases:

Case 1: $y_i^2 > y_j^2$**.** By equation 48, $Q > Q^\sigma$, so $R + Q > R + Q^\sigma$. Therefore $\mathbf{1}_{\mathcal{S}} \geq \mathbf{1}_{\mathcal{S}^\sigma}$ pointwise. The integrand is $(+)(+) \geq 0$ or $(+)(0) = 0$. ✓

Case 2: $y_i^2 < y_j^2$**.** By equation 48, $Q < Q^\sigma$, so $R + Q < R + Q^\sigma$. Therefore $\mathbf{1}_{\mathcal{S}} \leq \mathbf{1}_{\mathcal{S}^\sigma}$. The integrand is $(-)(-)$ or $(-)(0)$, both ≥ 0. ✓

Case 3: $y_i^2 = y_j^2$**.** The first factor is zero. ✓

In all cases, $(y_i^2 - y_j^2)\big(\mathbf{1}_{\mathcal{S}}(y) - \mathbf{1}_{\mathcal{S}^\sigma}(y)\big) \geq 0$ for every $y \in \mathbb{R}^p$. Substituting into equation 47:

$$2N = \mathbb{E}\Big[(y_i^2 - y_j^2)\big(\mathbf{1}_{\mathcal{S}} - \mathbf{1}_{\mathcal{S}^\sigma}\big)\Big] \geq 0,$$

since the integrand is pointwise non-negative and integrated against a probability measure. Hence $N \geq 0$, which gives $\mathbb{E}[y_i^2 \mid \mathcal{S}] \geq \mathbb{E}[y_j^2 \mid \mathcal{S}]$. □

A.5.3 Proof of Proposition 1 (Half-Space Expectation)

Proof. Let $u = \Sigma_{\text{filtered}}^{1/2} w$ where $w \sim \mathcal{N}(0, I)$, so $g_d \overset{d}{=} u$. The acceptance condition is $g_C^\top u > 0$, i.e., $g_C^\top \Sigma_{\text{filtered}}^{1/2} w > 0$, i.e., $a^\top w > 0$ where $a = \Sigma_{\text{filtered}}^{1/2} g_C$.

For a standard Gaussian w,

$$\mathbb{E}[w \mid a^\top w > 0] = \frac{a}{\|a\|}\, \mathbb{E}\left[\frac{a^\top w}{\|a\|} \;\middle|\; \frac{a^\top w}{\|a\|} > 0\right] = \frac{a}{\|a\|}\sqrt{\frac{2}{\pi}}, \tag{49}$$

where we used the fact that $a^\top w / \|a\| \sim \mathcal{N}(0,1)$ and $\mathbb{E}[|Z|] = \sqrt{2/\pi}$ for $Z \sim \mathcal{N}(0,1)$.

Transforming back:

$$\bar{g}_{\text{update}} = \Sigma_{\text{filtered}}^{1/2}\, \mathbb{E}[w \mid a^\top w > 0] = \sqrt{\frac{2}{\pi}}\, \frac{\Sigma_{\text{filtered}}^{1/2} a}{\|a\|} = \sqrt{\frac{2}{\pi}}\, \frac{\Sigma_{\text{filtered}}\, g_C}{\left\|\Sigma_{\text{filtered}}^{1/2} g_C\right\|}. \tag{50}$$

Since $\left\|\Sigma_{\text{filtered}}^{1/2} g_C\right\|^2 = g_C^\top \Sigma_{\text{filtered}}\, g_C$, the result follows. □

A.5.4 Proof of Theorem 8 (Spectral Whitening via Injection and Reward)

Proof. We trace the effect of each component on the condition number, restricting all eigenvalues and condition numbers to the support $\mathcal{S} = \text{range}(\Sigma_0)$.

Step 1: Random MoE injection (improves conditioning). By Theorem 6, $\Sigma_{\text{raw}} = \Sigma_0 + \varepsilon^2\, \Sigma_{\text{inject}}$. By Theorem 6(ii), both Σ_0 and Σ_{inject} are confined to $\mathcal{S}$. Restricting to $\mathcal{S}$ and applying the Weyl inequalities gives

$$\lambda_{\max}(\Sigma_{\text{raw}}|_{\mathcal{S}}) \leq \mu_1 + \varepsilon^2 \delta_1, \qquad \lambda_{\min}(\Sigma_{\text{raw}}|_{\mathcal{S}}) \geq \mu_s + \varepsilon^2 \delta_s. \tag{51}$$

Therefore, when $\Sigma_{\text{inject}}|_{\mathcal{S}} \succeq 0$ and $\delta_s/\delta_1 > \mu_s/\mu_1$,

$$\kappa^+(\Sigma_{\text{raw}}) \leq \frac{\mu_1 + \varepsilon^2\delta_1}{\mu_s + \varepsilon^2\delta_s} < \frac{\mu_1}{\mu_s} = \kappa^+(\Sigma_0). \tag{52}$$

Step 2: ReSTEM reward (improves conditioning). By Proposition 1, the effective covariance of accepted gradients, conditioned on the half-space $g_C^\top g_d > 0$, is the rank-1 downdate

$$\begin{aligned}\Sigma_{\text{dream}} &= \text{Cov}(g \mid g_C^\top g > 0) \\ &= \mathbb{E}[gg^\top \mid g_C^\top g > 0] - \mathbb{E}[g \mid g_C^\top g > 0]\,\mathbb{E}[g \mid g_C^\top g > 0]^\top \\ &= \Sigma_{\text{raw}} - \frac{2}{\pi}\,\frac{\Sigma_{\text{raw}}\, g_C\, g_C^\top\, \Sigma_{\text{raw}}}{g_C^\top \Sigma_{\text{raw}}\, g_C},\end{aligned} \tag{53}$$

where we used that $\mathbb{E}[gg^\top | g_C^\top g > 0] = \mathbb{E}[gg^\top]$ since the distribution of g is even. When g_C is aligned with the top eigenvector u_1 of $\Sigma_{\text{raw}}|_{\mathcal{S}}$ (as is typical since the task gradient concentrates along the dominant singular direction of the loss Hessian), the downdate selectively reduces the top eigenvalue, while leaving other eigenvalues unchanged. With $g_C = u_1$ and writing λ_1, λ_s for the extreme eigenvalues of $\Sigma_{\text{raw}}|_{\mathcal{S}}$: The downdate subtracts $(2/\pi)\,\lambda_1$ from the top eigenvalue and leaves all other eigenvalues unchanged. Therefore the post-downdate eigenvalues along u_1 and u_s are $(1 - 2/\pi)\lambda_1$ and λ_s, respectively. The new maximum eigenvalue is

$$\lambda_{\max}(\Sigma_{\text{dream}}|_{\mathcal{S}}) = \max\big((1 - \tfrac{2}{\pi})\lambda_1,\ \lambda_2\big), \qquad \lambda_{\min}(\Sigma_{\text{dream}}|_{\mathcal{S}}) = \lambda_s. \tag{54}$$

Since $\max((1-2/\pi)\lambda_1, \lambda_2) \leq \lambda_1$ (because both terms are $\leq \lambda_1$), we have $\kappa^+(\Sigma_{\text{dream}}) \leq \kappa^+(\Sigma_{\text{raw}})$, with strict inequality when $\lambda_1 > \lambda_2$. More precisely,

$$\kappa^+(\Sigma_{\text{dream}}) \leq \max\left(1 - \frac{2}{\pi},\ \frac{\lambda_2}{\lambda_1}\right) \kappa^+(\Sigma_{\text{raw}}). \tag{55}$$

When the spectral gap satisfies $\lambda_2/\lambda_1 \leq 1 - 2/\pi \approx 0.36$, this reduces to $\kappa^+(\Sigma_{\text{dream}}) \leq (1 - 2/\pi)\,\kappa^+(\Sigma_{\text{raw}})$.

Combining Steps 1–2. Chaining equation 52 and equation 55:

$$\kappa^+(\Sigma_{\text{dream}}) \;\leq\; \max\left(1 - \frac{2}{\pi},\ \frac{\lambda_2}{\lambda_1}\right) \cdot \underbrace{\frac{\mu_1 + \varepsilon^2\delta_1}{\mu_s + \varepsilon^2\delta_s}}_{< \mu_1/\mu_s} \;<\; \frac{\mu_1}{\mu_s} \;=\; \kappa^+(\Sigma_0), \tag{56}$$

which is equation 31. When the spectral gap satisfies $\lambda_2/\lambda_1 \leq 1-2/\pi$, the stronger bound equation 32 holds.

Extension with filtering. If gradient filtering (Theorem 7) is applied between injection and ReSTEM, the filtered covariance has the same eigenbasis as Σ_{raw} with amplified eigenvalues $\tilde{\lambda}_i = r_i\,\lambda_i$. By Part (iii) of Theorem 7, the amplification ratio r_i is non-decreasing in λ_i, so $r_1 \geq r_p$ and

$$\kappa^+(\Sigma_{\text{filtered}}) = \frac{r_1}{r_p}\,\kappa^+(\Sigma_{\text{raw}}) \geq \kappa^+(\Sigma_{\text{raw}}). \tag{57}$$

Applying the ReSTEM step to Σ_{filtered} instead of Σ_{raw}, cf. equation 55 gives

$$\kappa^+(\Sigma_{\text{dream}}) \leq \max\left(1 - \frac{2}{\pi},\ \frac{\lambda_2}{\lambda_1}\right) \frac{r_1}{r_p}\,\kappa^+(\Sigma_{\text{raw}}),$$

which yields equation 33, by using equation 52. □

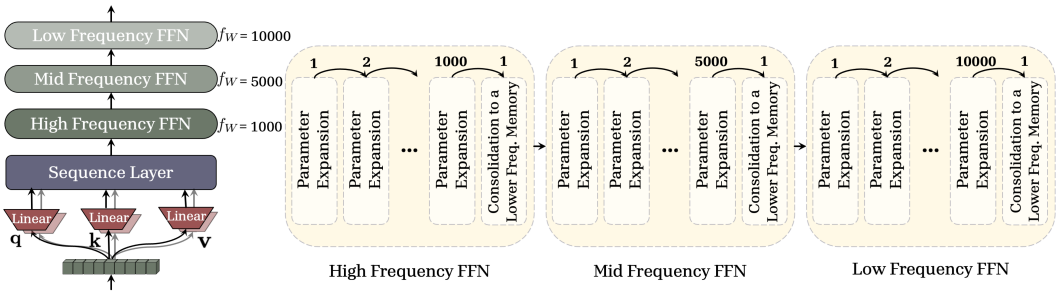


Figure 6: Multi-frequency memory hierarchy. Updates enter the High-Frequency FFN via repeated Parameter Expansion; when the window f_W expires, knowledge is Consolidated to the Mid- and then Low-Frequency FFNs (1k→5k→10k).

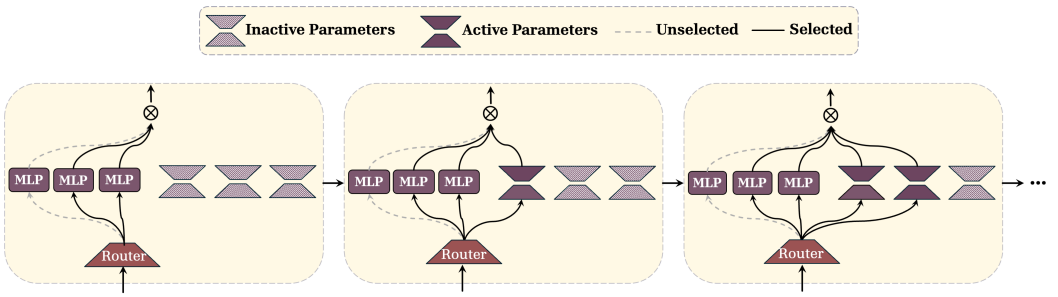


Figure 7: Memory consolidation by routed expert updates. Across Sleep cycles (left→right), a router selects and updates a small set of experts (solid), leaving others inactive (hatched), expanding capacity while limiting interference.

B Related Work

B.1 Parameter-Efficient Adaptation and Composition

Parameter-efficient fine-tuning (PEFT) adapts large language models (LLMs) to specific tasks by optimizing a minimal set of auxiliary parameters while freezing the backbone. **Low-Rank and Prefix Adaptation.** Prominent methods include LoRA (Hu et al., 2022), which injects trainable low-rank matrices into linear projections, and prefix/prompt-tuning (Li & Liang, 2021; Lester et al., 2021), which steers computation via learnable virtual tokens. Recent variants optimize initialization to accelerate convergence, such as PiSSA (Meng et al., 2024), or extend these mechanisms for context-to-parameter distillation (Eyuboglu et al., 2025).

Composition and Routing. Beyond single-task adaptation, research increasingly focuses on composing multiple adapters. Approaches include weighted composition via in-context learning (Huang et al., 2023) and retrieval-based routing, where relevant LoRA modules are selected dynamically per input (Zhao et al., 2024a). Advanced configurations utilize mixture-of-experts (MoE) architectures to activate specific low-rank paths or merge adapter parameters dependent on the task (Xiao et al., 2024; Yadav et al., 2024; Wu et al., 2024; Gou et al., 2023; Li et al., 2024a; Zhao et al., 2024b).

B.2 Knowledge Injection, Distillation, and Self-Improvement

Parametric Knowledge Injection. To reduce reliance on retrieval at inference time, recent work injects external knowledge directly into model parameters. Techniques range from *Parametric RAG*, which assigns document-specific LoRA adapters (Su et al., 2025), to prompt distillation, where student models learn from teacher-generated QA pairs or synthetic conversations (Kujanpää et al., 2024; Mao et al., 2025; Caccia et al., 2025). *Cartridges* (Eyuboglu et al., 2025) bridge PEFT and

distillation by pre-training a reusable "KV cache adapter" via a self-study objective, achieving in-context learning (ICL) quality with reduced serving costs.

Self-Improvement and Meta-Learning. Complementary to distillation is the use of model-generated signals for self-improvement. This includes Reinforcement Learning (RL) on verifiable rewards to enhance reasoning (DeepSeek-AI, 2025; Singh et al., 2024b; Zelikman et al., 2022) and self-rewarding mechanisms (Wang et al., 2025; Pang et al., 2024b). Meta-learning frameworks, such as SEAL (Zweiger et al., 2025), extend this by optimizing the adaptation strategy itself—learning *how* to generate self-edits—drawing on broader principles of self-referential learning (Schmidhuber, 1987; Finn et al., 2017; Irie et al., 2022). These methods rely heavily on high-quality synthetic data generation to scale supervision (Nayak et al., 2024; Abdin et al., 2024; Riaz et al., 2025).

A recent group of concurrent studies has suggested to use on-policy distillation for self-distillation process (Zhao et al., 2026a; Hübotter et al., 2026; Zhang et al., 2026b) or for continual learning (Shenfeld et al., 2026a). In addition to the fact that this study is older than such methods, Sleep fundamentally delivers a different messages that: (1) Sleep uses an *upward distillation of self*, where it unlocks parameters. (2) Sleep is based on Generalized Distillation method that combines on-policy method with an RL method. (3) Sleep suggest a periodic process where knowledge stores in modules in different frequency of update.

B.3 Efficient Context Processing

Addressing the memory bottleneck of long-context processing involves compressing the KV cache or modifying the attention architecture.

Prompt and Cache Compression. Approaches to reduce memory footprints fall into two categories: *Prompt compression* shortens inputs via token filtering (hard-token) (Li, 2023; Jiang et al., 2023) or by learning compact soft-token embeddings via auto-encoders (Chevalier et al., 2023; Ge et al., 2023; Tan et al., 2024). *KV cache compression* operates at runtime, employing eviction policies to drop non-essential keys (Tang et al., 2024; Zhang et al., 2023; Li et al., 2024b) or merging tokens based on similarity and attention density (Wan et al., 2024; Liu et al., 2024; Zhang et al., 2024b; Wang et al., 2024b). Low-rank projections of the KV cache have also shown promise in maintaining performance at high compression rates (Zhang et al., 2024a; Chari et al., 2025).

Architectural Innovations. Structural changes to Multi-Head Attention (MHA) include Multi-Query and Grouped-Query Attention (MQA/GQA), which share KV heads to reduce memory bandwidth (Shazeer, 2019; Ainslie et al., 2023), and linearization techniques that remove the softmax to achieve fixed-size states ($K^\top V$) independent of sequence length (Arora et al., 2024; Gu & Dao, 2023; Beck et al., 2024). Recent "learning to compress" architectures, such as Titans and TTT, utilize gradient-based updates on constant-sized memory objects to handle infinite contexts (Behrouz et al., 2024; Sun et al., 2024). Finally, orchestration systems like MemGPT manage context via virtual memory paging rather than architectural modification (Packer et al., 2023). In a similar direction to compression in the token space, Lin et al. (2025) present sleep-time compute. Despite similarity in the name, their method is fundamentally different from our work. While this method aim to find a good summary of the interactions of the model with users in the text space, when the model is idle, our proposal is on using distillation to transfer text data knowledge into a form of parametric weight update.

B.4 Recent Concurrent and/or Later Work on On-Policy Distillation

Concurrent and Subsequent Work on On-Policy Self-Distillation. Concurrent and subsequent to the initial version of this work, a rapidly growing line of literature has explored *on-policy self-distillation* (OPSD), in which the same model plays the role of both teacher and student under different conditioning contexts and the student is supervised on its own rollouts via a per-token divergence against the teacher. The canonical instantiation is OPSD (Zhao et al., 2026b), which conditions the teacher on a verified reasoning trace and minimises a per-token reverse-KL on the student's own rollouts, yielding gains on mathematical reasoning over both GRPO and off-policy distillation. A wave of follow-up works varies the form of the privileged context: On-Policy Context Distillation (Ye et al., 2026b) conditions the teacher on transient in-context information (historical solution traces or optimised system prompts) so that the student internalises that information into its parameters; GATES (Stein et al., 2026) uses document-grounded privileged context together with a consensus-

gating mechanism that handles unreliable supervision by sampling multiple tutor traces and gating learning by their agreement; SD-Zero (He et al., 2026) conditions a "reviser" on the student's response and a binary reward, then distils the reviser back into the generator, converting sparse outcome rewards into dense token-level supervision; and COPSD (Liu et al., 2026) transfers reasoning behaviour to low-resource languages by giving the teacher an English translation and reference solution as privileged crosslingual context. Apple's "embarrassingly simple" self-distillation (Zhang et al., 2026a) sits at the limit of this spectrum: it drops the explicit teacher altogether and supervised-finetunes on the model's own samples under different decoding configurations, showing that even this minimal recipe meaningfully improves code generation.

Continual, Experiential, and Online Adaptation. A second cluster of works applies self-distillation to continual, experiential, or online improvement. SDFT (Shenfeld et al., 2026b) explicitly targets continual learning from demonstrations by using a demonstration-conditioned model as an on-policy teacher, reducing catastrophic forgetting compared with standard SFT. OEL (Ye et al., 2026a) couples context distillation with deployment: it extracts transferable experiential knowledge from interaction trajectories and consolidates it into parameters via on-policy context distillation, iterating both stages to form a closed online-learning loop. π-Play (Zhang et al., 2026d) pushes this idea further into a multi-agent self-play regime, using the question-construction path produced by the task proposer as privileged context for the teacher and removing the need for external data or human feedback in training deep search agents. These works are closely aligned with our motivation that LLMs should not remain static after deployment. However, they formulate continual or experiential improvement as a post-training objective on a *fixed-capacity* checkpoint. In contrast, our Sleep paradigm treats continual learning as a *memory-system* problem: newly acquired information is first represented in fast, fragile memory modules and then periodically consolidated into slower, more stable parameters via Knowledge Seeding, accompanied by gradual capacity expansion and the resetting of higher-frequency memory after consolidation.

Application-Specific OPSD Recipes. A third group of papers adapts the OPSD template to specific deployment regimes. CRISP/OPSDC (Sang et al., 2026) conditions the teacher on a "be concise" instruction prefix and uses per-token reverse-KL on student rollouts to compress long chain-of-thought traces without entropy collapse. MTP (Kirchenbauer et al., 2026) converts a pretrained next-token-prediction model into a multi-token predictor with a single online self-distillation objective, yielding more than $3\times$ inference acceleration on GSM8K at $< 5\%$ accuracy drop. OPSDL (Zhang et al., 2026c) attacks long-context generation by using the model's own short-context behaviour on extracted relevant spans as a self-teacher for its long-context student. Skill-SD (Wang et al., 2026) targets multi-turn agents: completed trajectories are summarised into compact natural-language "skills" that condition only the teacher while the student trains under the plain task prompt. MSD (Qin et al., 2026) moves OPSD into the multilingual setting by transferring safety behaviour from high-resource to low-resource languages using only multilingual queries. Collectively, these works show that self-distillation under privileged conditioning is a general and flexible post-training recipe across reasoning, efficiency, agentic, and multilingual axes – but in each case it is deployed as a single-objective procedure on a fixed model and a fixed application.

Limitations of OPSD and Motivation for the Sleep Paradigm. Despite the empirical success of OPSD across these domains, recent analyses have begun to expose its failure modes. Kim et al. (2026) show that, in mathematical reasoning, conditioning the teacher on rich privileged information can suppress the model's epistemic verbalisation (its expression of uncertainty during reasoning), enabling fast in-domain optimisation at the cost of severe out-of-distribution degradation, with drops of up to 40% across Qwen3-8B, DeepSeek-Distill-Qwen-7B, and Olmo3-7B-Instruct. Several recent OPSD works also report that naïvely iterating self-distillation in a long-horizon training pipeline can lead to information leakage from the privileged teacher and to training collapse (Wang et al., 2026; He et al., 2026). These results suggest that self-distillation alone is not sufficient for robust continual improvement: an iterative self-improvement loop applied directly to a fixed model can erode prior capabilities or destabilise reasoning. Our two-stage Sleep framework directly addresses this concern by separating memory consolidation from self-improvement: consolidation first stabilises newly acquired knowledge in freshly expanded lower-frequency parameters before Dreaming further modifies the model, reducing the risk that iterative self-training overwrites useful prior capabilities.

Positioning of Our Work. Our work differs from the OPSD literature along four axes that together motivate the Sleep paradigm. **(i) Upward distillation as memory consolidation.** All of the self-distillation works listed above share the same backbone between teacher and student and differ only in conditioning context – privileged trace (Zhao et al., 2026b; Kim et al., 2026), demonstration (Shenfeld et al., 2026b), document (Stein et al., 2026), context or system prompt (Ye et al., 2026b;a), "concise" prefix (Sang et al., 2026), decoding configuration (Zhang et al., 2026a), skill summary (Wang et al., 2026), reviser context (He et al., 2026), question-construction path (Zhang et al., 2026d), short-context substring (Zhang et al., 2026c), or English reference (Qin et al., 2026; Liu et al., 2026). In contrast, our Knowledge Seeding step is an *upward* distillation: a smaller, higher-frequency memory module is distilled into a strictly *larger* set of newly expanded low-rank experts in a lower-frequency memory module. This reframes catastrophic forgetting as a problem of insufficient capacity rather than of sampling distribution, and addresses it by gradual parameter growth between consolidation steps rather than by re-conditioning a fixed teacher. **(ii) Continuum of memory frequencies.** Where the above works treat distillation as a flat student/teacher pair, our method operates over a chain of memory blocks with strictly ordered update frequencies and performs consolidation between each pair of consecutive blocks; to our knowledge, only SDFT (Shenfeld et al., 2026b) explicitly targets the continual-learning desideratum that motivates us, and it does so without architectural growth or hierarchical memory. **(iii) Sleep as a two-phase process beyond consolidation.** The OPSD literature focuses almost entirely on the consolidation step. Our framework additionally includes a Dreaming phase analogous to REM sleep, in which the model generates a curriculum of synthetic data, weights samples by a gradient-based importance score, and exploits the MoE router to inject controlled novelty – explicitly designed to be robust to the iterative self-improvement failure modes flagged by (Kim et al., 2026; He et al., 2026). **(iv) Knowledge seeding augments on-policy distillation with imitation learning.** Where the on-policy distillation objective in works such as (Zhao et al., 2026b; Ye et al., 2026b; Stein et al., 2026; Sang et al., 2026; Ye et al., 2026a; Zhang et al., 2026c; Qin et al., 2026; Liu et al., 2026) reduces to a per-token reverse-KL on student rollouts, our Knowledge Seeding objective augments GKD-style on-policy distillation with an RL-based Learning-to-Imitate term that jointly rewards semantic and Levenshtein-level alignment with the teacher's sampling distribution, enabling the larger student not only to inherit the teacher's knowledge but also to imitate the way the teacher uses it.

C PSEUDOCODE

Algorithm 1 WAKE–SLEEP LIFECYCLE

Input: $\mathrm{LM}_{\boldsymbol{\theta}}$ with sequence model + CMS chain $\mathrm{MLP}^{(f_1)}, \ldots, \mathrm{MLP}^{(f_k)}$; chunk sizes $\{C^{(1)}, \ldots, C^{(k)}\}$ with $C^{(\ell-1)} \mid C^{(\ell)}$; learning rates $\eta_t^{(\ell)}$; input stream $x_1, x_2, \ldots$

for $i = 1, 2, \ldots$ **do**

Active (wake) phase

  Receive x_i; compute $y_i = \mathrm{MLP}^{(f_k)}\big(\mathrm{MLP}^{(f_{k-1})}(\cdots \mathrm{MLP}^{(f_1)}(x_i))\big)$ ▷ Eq. 1; online consolidation

Sleep phase (triggered only at $i \in \{C^{(\ell)} \times b : b \in \mathbb{N}\}$)

  if $\exists\, \ell : i \equiv 0 \pmod{C^{(\ell)}}$ **then**

    for $\ell^* = k, k-1, \ldots, 2$ **such that** $i \equiv 0 \pmod{C^{(\ell^*-1)}}$ **do** ▷ slowest first

      $\mathrm{LM}_{\boldsymbol{\theta}} \leftarrow$ current state of the model ▷ teacher (pre-update $\mathrm{MLP}^{(f_{\ell^*-1})}$)

      ParameterExpansion$(\mathrm{MLP}^{(f_{\ell^*})})$ ▷ Algorithm 2

      $\boldsymbol{e}_{i,\ell^*-1} \leftarrow \sum_{t=i-C^{(\ell^*-1)}}^{i} \eta_t^{(\ell^*-1)}\, f\big(\boldsymbol{\theta}_t^{(f_{\ell^*-1})}; x_t\big)$ ▷ Eq. 2

      $\boldsymbol{\theta}^{(f_{\ell^*-1})} \leftarrow \boldsymbol{\theta}^{(f_{\ell^*-1})} - \boldsymbol{e}_{i,\ell^*-1}$ ▷ block $\ell^* - 1$ is updated

      $\mathrm{LM}_{\boldsymbol{\theta}_{\mathrm{exp}}} \leftarrow$ current (expanded and updated) state of the model ▷ student

      KnowledgeSeeding$(\mathrm{LM}_{\boldsymbol{\theta}}, \mathrm{LM}_{\boldsymbol{\theta}_{\mathrm{exp}}}, \ell^*)$ ▷ Algorithm 3: NREM-like consolidation

      Reset all low-rank experts $\{\boldsymbol{A}^{(f_{\ell^*-1}),j}, \boldsymbol{B}^{(f_{\ell^*-1}),j}\}$ previously added to $\mathrm{MLP}^{(f_{\ell^*-1})}$ ▷ pruning

    end for

    $\boldsymbol{\theta} \leftarrow$ Dreaming$(\mathrm{LM}_{\boldsymbol{\theta}})$ ▷ Algorithm 5: REM-like self-improvement

  end if

end for

Algorithm 2 PARAMETEREXPANSION: activate a new low-rank expert in the receiving block $\mathrm{MLP}^{(f_{\ell^*})}$

Input: Sparse-MoE block $\mathrm{MLP}^{(f_{\ell^*})}$ with router $\mathcal{R}^{(f_{\ell^*})}$ and experts $\{\boldsymbol{W}^{(f_{\ell^*}),1}, \ldots, \boldsymbol{W}^{(f_{\ell^*}),s_{\ell^*}}\}$; width d; low rank $d_{\mathrm{low}} \ll d$

Output: Expanded block with $s_{\ell^*}+1$ experts

1: Initialize $\boldsymbol{A}^{(f_{\ell^*}),s_{\ell^*}+1} \in \mathbb{R}^{d \times d_{\mathrm{low}}}$, $\boldsymbol{B}^{(f_{\ell^*}),s_{\ell^*}+1} \in \mathbb{R}^{d_{\mathrm{low}} \times d}$

2: Define the new expert $\boldsymbol{W}^{(f_{\ell^*}),s_{\ell^*}+1}(\boldsymbol{h}) := \boldsymbol{A}^{(f_{\ell^*}),s_{\ell^*}+1}\boldsymbol{B}^{(f_{\ell^*}),s_{\ell^*}+1}\boldsymbol{h}$ ▷ low-rank MLP

3: $\{\boldsymbol{W}^{(f_{\ell^*}),j}\}_{j=1}^{s_{\ell^*}} \leftarrow \{\boldsymbol{W}^{(f_{\ell^*}),j}\}_{j=1}^{s_{\ell^*}} \cup \{\boldsymbol{W}^{(f_{\ell^*}),s_{\ell^*}+1}\}$

4: Extend the output of $\mathcal{R}^{(f_{\ell^*})}$ to route over $s_{\ell^*}+1$ experts

5: $s_{\ell^*} \leftarrow s_{\ell^*}+1$

6: Mark $\{\boldsymbol{A}^{(f_{\ell^*}),s_{\ell^*}}, \boldsymbol{B}^{(f_{\ell^*}),s_{\ell^*}}\}$ as the *only* trainable parameters for the upcoming Knowledge Seeding

Implementation note (Sec. 3.3): instead of growing tensors, all low-rank experts may be pre-allocated and *masked* in forward/backward passes; this step then simply un-masks the next one.

Algorithm 3 KNOWLEDGESEEDING (self-Knowledge Seeding, SKS): upward distillation $\mathrm{LM}_{\boldsymbol{\theta}} \rightarrow \mathrm{LM}_{\boldsymbol{\theta}_{\mathrm{exp}}}$ via GKD + Learning to Imitate

Input: Teacher $\mathrm{LM}_{\boldsymbol{\theta}}$ (smaller, pre-expansion); student $\mathrm{LM}_{\boldsymbol{\theta}_{\mathrm{exp}}}$ (larger, expanded); receiving block index ℓ^*; on-policy fraction $\lambda \in [0,1]$; distillation weight $\alpha \in [0,1]$; reward mix $\gamma \in [0,1]$; threshold z_0; number of LTI sequences n; divergence $\mathcal{F}(\cdot\|\cdot)$; number of iterations T_{KS}; step size η

Output: Updated expanded parameters of $\mathrm{MLP}^{(f_{\ell^*})}$; all other parameters unchanged

(a) Build the sleep-time datasets from the teacher (no external data)

$\mathcal{D} \leftarrow \{(x,y) : y \sim \mathrm{LM}_{\boldsymbol{\theta}}(\cdot \mid x)\}$ ▷ teacher-generated (prompt, continuation) pairs

$\mathcal{D}_T \leftarrow \{d^{(1)}, \ldots, d^{(n)}\}, \quad d^{(i)} \sim \mathrm{LM}_{\boldsymbol{\theta}}(\cdot)$ ▷ teacher-generated "dreams" for LTI

(b) Freeze everything except the new expert

$\Delta \leftarrow \{\boldsymbol{A}^{(f_{\ell^*}),s_{\ell^*}}, \boldsymbol{B}^{(f_{\ell^*}),s_{\ell^*}}\}$; freeze $\boldsymbol{\theta}_{\mathrm{exp}} \setminus \Delta$ ▷ no interference with stored knowledge

(c) Optimize the Knowledge Seeding objective

for $t = 1, \ldots, T_{\mathrm{KS}}$ **do**

  Sample a minibatch $\{(x,y)\} \subset \mathcal{D}$

  Generalized (on-policy) distillation term

  With probability λ: $y \leftarrow \mathrm{sg}\big[\, y' \sim \mathrm{LM}_{\boldsymbol{\theta}_{\mathrm{exp}}}(\cdot \mid x)\big]$ ▷ on-policy; no backprop through student sampling

  otherwise keep the teacher's y ▷ off-policy share $(1-\lambda)$

  $\mathcal{L}_{\mathrm{dist}} \leftarrow \mathcal{F}(\mathrm{LM}_{\boldsymbol{\theta}} \,\|\, \mathrm{LM}_{\boldsymbol{\theta}_{\mathrm{exp}}})(y \mid x)$ ▷ token-level feedback from teacher logits

  Learning-to-Imitate (LTI) term

  Sample $d^{(i)} \in \mathcal{D}_T$; sample a prefix length p and set $d^{(i)}_{\leq p}$ ▷ teacher masks the remainder

  $\hat{d}^{(i)} \sim \mathrm{LM}_{\boldsymbol{\theta}_{\mathrm{exp}}}(\cdot \mid d^{(i)}_{\leq p})$ ▷ student completes the continuation

  $r \leftarrow r\big(\hat{d}^{(i)}; d^{(i)}; \mathrm{LM}_{\boldsymbol{\theta}_{\mathrm{exp}}}\big)$ ▷ Algorithm 4, Eq. 3–4

  Combined objective (maximized)

  $\mathcal{L}_{\mathrm{KS}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{\mathrm{exp}}) \leftarrow (1-\alpha)\, r \;-\; \alpha\, \mathcal{L}_{\mathrm{dist}}$

  population form: $\mathcal{L}_{\mathrm{KS}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{\mathrm{exp}}) = \mathbb{E}_{x \sim \mathcal{D}}\Big[(1-\alpha)\, \mathbb{E}_{y \sim \mathrm{LM}_{\boldsymbol{\theta}_{\mathrm{exp}}}(\cdot|x)}[r(y)] - \alpha\, \mathbb{E}_{y \sim \mathrm{LM}_{\boldsymbol{\theta}_{\mathrm{exp}}}(\cdot|x)}\big[D(\mathrm{LM}_{\boldsymbol{\theta}} \,\|\, \mathrm{LM}_{\boldsymbol{\theta}_{\mathrm{exp}}})(y|x)\big]\Big]$

  $\Delta \leftarrow \Delta + \eta\, \widehat{\nabla}_{\Delta} \mathcal{L}_{\mathrm{KS}}$ ▷ PG on reward term; exact grad on divergence

end for

return $\mathrm{LM}_{\boldsymbol{\theta}_{\mathrm{exp}}}$ with updated Δ ▷ memory of $\mathrm{MLP}^{(f_{\ell^*-1})}$ now consolidated into $\mathrm{MLP}^{(f_{\ell^*})}$

Algorithm 4 LTI REWARD $r(\hat{d}^{(i)};\, d^{(i)};\, \mathrm{LM}_{\boldsymbol{\theta}_{\mathrm{exp}}})$

Input: Student completion $\hat{d}^{(i)}$; teacher sequence $d^{(i)}$; frozen semantic reward model RM; Levenshtein distance $z(\cdot,\cdot)$; threshold z_0; mix γ

Semantic reward

$r_{\mathrm{sem}} \leftarrow \begin{cases} 1 & \text{if RM judges the semantics of } \hat{d}^{(i)} \text{ and } d^{(i)} \text{ to be the same} \\ 0 & \text{otherwise} \end{cases}$

Absolute (token-level) reward

$r_{\mathrm{abs}} \leftarrow \begin{cases} 1 - \dfrac{z(\hat{d}^{(i)}, d^{(i)})}{\max\{|\hat{d}^{(i)}|,\, |d^{(i)}|\}} & \text{if } z(\hat{d}^{(i)}, d^{(i)}) \le z_0 \\ 0 & \text{otherwise} \end{cases}$ ▷ Eq. 4

return $r \leftarrow \gamma\, r_{\mathrm{sem}} + (1-\gamma)\, r_{\mathrm{abs}}$ ▷ Eq. 3

Algorithm 5 DREAMING Stage

Input: Post-consolidation model $\mathrm{LM}_{\boldsymbol{\theta}}$ (MoE blocks with routers $\mathcal{R}^{(f_\ell)}$); task distribution over (C, τ), where C is a context and $\tau(\cdot)$ a downstream performance measure; number of dreams $m \ge 1$; Top-k; b random picks; LM objective $\mathcal{L}_{SFT}$; number of $\mathrm{ReST}^{\mathrm{EM}}$ rounds E

Output: Self-modified parameters $\boldsymbol{\theta}$

for $e = 1, \ldots, E$ **do** ▷ one $\mathrm{ReST}^{\mathrm{EM}}$ outer iteration
  $\mathcal{B} \leftarrow \emptyset$ ▷ buffer of successful dreams
  for each sampled task (C, τ) **do**
  (a) Generate dreams with exploratory routing
    for $i = 1, \ldots, m$ **do**
      In every MoE block $\mathrm{MLP}^{(f_\ell)}$, let $\mathcal{R}^{(f_\ell)}$ pick its usual experts **plus one random expert** ▷ novel synthesis
      $\mathrm{DREAM}^{(i)} \sim \mathrm{LM}_{\boldsymbol{\theta}}(\cdot \mid C)$
    end for
  (b) Gradient-based rejection / selection
    for $i = 1, \ldots, m$ **do**
      $g_{\mathrm{DR}}^{(i)} \leftarrow \nabla_{\boldsymbol{\theta}}\, \mathcal{L}_{SFT}(\mathrm{DREAM}^{(i)}, \boldsymbol{\theta})$ ▷ importance score defined by the gradient
      $\boldsymbol{\omega}^{(i)} \leftarrow \mathrm{score}\big(g_{\mathrm{DR}}^{(i)}\big)$ ▷ scalarization used for ranking, e.g. $\|g_{\mathrm{DR}}^{(i)}\|$
    end for
    $\mathcal{D}_{\mathrm{DR}} \leftarrow \text{Top-}k\big(\{\mathrm{DREAM}^{(i)}\}_{i=1}^{m};\, \boldsymbol{\omega}\big) \,\cup\, \mathrm{Random}_b\big(\text{remaining dreams}\big)$ ▷ Top-$k + b$ random
  (c) Isolated inner-loop adaptation and binary reward (Eq. 5)
    for each $\mathrm{DREAM}^{(i)} \in \mathcal{D}_{\mathrm{DR}}$ **do**
      $\boldsymbol{\theta}^{(i)} \leftarrow \boldsymbol{\theta}$ ▷ isolated instance of the model
      $\boldsymbol{\theta}'^{(i)} \leftarrow \mathrm{SFT}\big(\boldsymbol{\theta}^{(i)}, \mathrm{DREAM}^{(i)}\big)$ ▷ supervised fine-tuning with LoRA
      $r\big(\mathrm{DREAM}^{(i)}, \tau(\cdot), \mathrm{LM}_{\boldsymbol{\theta}^{(i)}}\big) \leftarrow \begin{cases} 1 & \text{if } \tau\big(\mathrm{LM}_{\boldsymbol{\theta}'^{(i)}}\big) \text{ improves over } \tau\big(\mathrm{LM}_{\boldsymbol{\theta}^{(i)}}\big) \\ 0 & \text{otherwise} \end{cases}$
      if $r = 1$ **then** $\mathcal{B} \leftarrow \mathcal{B} \cup \{(C, \mathrm{DREAM}^{(i)})\}$
      end if
    end for
  end for
  (d) $\mathrm{ReST}^{\mathrm{EM}}$ M-step: reinforce the dream-generation policy
  $\boldsymbol{\theta} \leftarrow \mathrm{SFT}\big(\boldsymbol{\theta},\, \mathcal{B}\big)$ ▷ SFT on rewarded dreams
end for
return $\boldsymbol{\theta}$

D Additional Experimental Results and Details

In all of our experiments, we follow the settings of the original benchmark (cited in each section). Therefore, for the sake of space and not duplicating information, we refer to those studies for the details of the models. In our design, we use 5 MLP blocks with dimension 64 as the additional parameters and keep the active parameter count unchanged (I.e., the same as the base model, which is either 8B or 3B).

D.1 Datasets

Class Incremental Learning. We focus on class-incremental learning on three datasets:

- CLINC (Larson et al., 2019): CLINC150 is a multi-domain intent classification benchmark for task-oriented dialog, and it is commonly used to evaluate both in-scope intent prediction and out-of-scope (OOS) detection. It contains 150 in-scope intents spanning 10 domains, with 23.7K total queries (22.5K in-scope and 1.2K OOS).
- Banking (Casanueva et al., 2020): Banking77 is a single-domain intent dataset of short banking customer-service queries, labeled with 77 fine-grained intents (e.g., card issues or PIN resets). It includes 13,083 examples, and the intent distribution is noticeably imbalanced.
- DBpedia (Auer et al., 2007): DBpedia-based classification maps Wikipedia-derived descriptions (typically abstracts) to ontology categories (e.g., book, film, animal, place). We use the 70-class, level-2 setting and subsample 10K training and 1K test instances for our experiments.

The Effect of Levels on In-context Learning. To better isolate how Hope's consolidation schedule impacts in-context learning and long-context understanding, we evaluate question answering and multi-key retrieval under long contexts. We use the following datasets for this evaluation:

- LongHealth (Adams et al., 2025): LongHealth is a long-context clinical multiple-choice QA benchmark built from lengthy fictional patient-case records. It includes 20 case documents (about 5.1K–6.8K words each); we evaluate on 200 questions sampled from these records.
- QASPER (Dasigi et al., 2021): QASPER is an information-seeking QA benchmark grounded in full-text NLP papers, with roughly 5K questions over about 1.6K papers. We use each paper's full text as the context.
- MK-NIAH (Hsieh et al., 2024): MK-NIAH is the multi-key needle-in-a-haystack task from RULER (Hsieh et al., 2024), where multiple key–value facts are embedded in a long context and the model must retrieve the value for a queried key.

D.2 Additional Results: BABILong

We evaluate Hope (Sleep) on the BABILong benchmark (Kuratov et al., 2024), comparing against: (1) large models (GPT-4 and GPT-4o-mini (Achiam et al., 2023)); (2) a mid-scale Llama-8B model (Dubey et al., 2024) with RAG; and (3) state-of-the-art small long-context models, including RMT (Bulatov et al., 2022), ARMT (Rodkin et al., 2024), and Titans (Behrouz et al., 2024). All small models are fine-tuned using the official BABILong training protocol (Kuratov et al., 2024). As shown in Figure 5, large models exhibit rapid degradation as context length increases, failing beyond 128K–256K tokens. RAG improves robustness at longer contexts but still degrades as sequence length grows. Among fine-tuned small models, Titans, ARMT, and Hope perform comparably up to approximately 1M tokens; beyond this point, Titans and ARMT degrade sharply, whereas Hope remains stable even at extreme lengths up to 10M tokens. This robustness is driven by Sleep's explicit consolidation and self-improvement mechanism, which transforms short-lived activations into compact parametric representations. By learning higher-level abstractions during consolidation, Hope retains task-relevant information more efficiently, reducing sensitivity to increasing context length. Finally, we observe that all small models, including Hope, suffer substantial performance drops without fine-tuning. Compressing ultra-long contexts requires sufficient capacity at lower-frequency memory levels to identify and preserve critical information; fine-tuning enables these components to adapt rapidly and coordinate effectively with high-frequency memory during inference.

D.3 Additional Experiments: ARC

We follow the few-shot ARC experimental protocol from prior work (Akyürek et al., 2024a; Zweiger et al., 2025) and adapting it to our SLEEP paradigm. As the backbone we use `Llama-3.2-1B`. Following common practice, we filter subset of data to avoid tasks that remain unsolvable under standard configurations, yielding 11 tasks for training and 8 held-out tasks for evaluation. During each *Sleep* cycle, the model first consolidate its previous memories, and then *dreams* by generating synthetic experiences from the few-shot demos. For each task, we sample 60 dreams and reject 45 of them. At test time, for each unseen task, the model generates 5 dreams and applies them independently before predicting the held-out output. We report the fraction of dreams that yield a correct answer. As the baselines, we follow Zweiger et al. (2025) and use: (i) ICL (In-Context Learning); (ii) TTT + synthetic updates (no dreaming); and (iii) SEAL (Zweiger et al., 2025). In this setting, SLEEP achieves a 80% success rate, higher then the other methods.

D.4 Additional Details

Table 2: Training Configuration for GRPO, SFT, and Sleep

Parameter	**GRPO**	**SFT**	**Sleep**
Learning Rate	5×10^{-6}	5×10^{-6}	5×10^{-6}
Effective Batch Size	32	32	32
Training Steps	500	100	100
LoRA Rank (r)	64	64	64
LoRA Alpha (α)	128	128	128

D.5 Efficiency

With the same number of steps, SFT is 4x more efficient than our method, however, their performance is not comparable. Accordingly, we also compare the efficiency, when targeting a specific performance. We trained the models so they achieve the same performance in AIME-24, AIME-25, and HMMT-25. In this case, SFT requires 4.3x, 3.6x, and 4.8x wall-clock time to match the performance of our design. Therefore, from this perspective, Sleep is very efficient, when targeting a specific performance.

E Additional Experimental Results

This section contains additional figures and tables referenced from the main text.

E.1 Effect of Memory Levels on In-Context Learning

Figure 8 presents the full results for the effect of the number of memory levels (sleep phases) on in-context learning performance. We evaluate on three benchmarks: MK-NIAH from RULER (Hsieh et al., 2024), LongHealth (Adams et al., 2025), and QASPER (Dasigi et al., 2021). The results show that HOPE consistently outperforms all baselines (ICL, DuoAttention (Xiao et al., 2025), and Cartridges (Eyuboglu et al., 2025)). Increasing the number of consolidation stages generally improves performance, while increasing the lowest frequency weakens retention.

E.2 Mathematical Reasoning

Table 3 presents the full mathematical reasoning results across both Qwen3-1.7B and Qwen3-8B model sizes. We compare against base models (Instruct), SFT, GRPO (Shao et al., 2024), and OPSD. Sleep consistently outperforms all baselines across all benchmarks and model sizes.

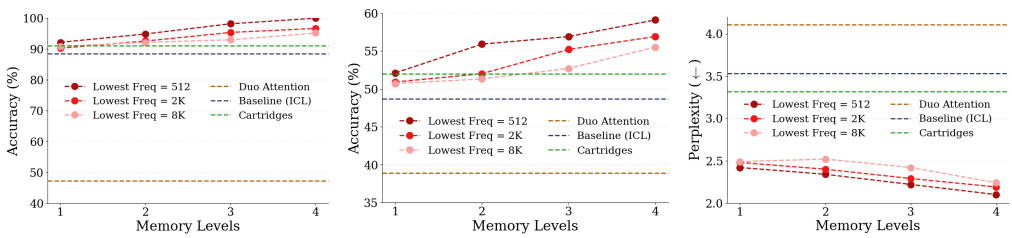


Figure 8: Effect of memory levels on in-context learning performance for (Left) MK-NIAH from RULER (Hsieh et al., 2024), (Middle) LongHealth (Adams et al., 2025), and (Right) QASPER (Dasigi et al., 2021). Lower values indicate better performance for QASPER.

Table 3: Performance of different methods on mathematical reasoning benchmarks. We use different variants of Qwen models and report average@16.

Method	**AIME-24**	**AIME-25**	**HMMT-25**
	Qwen3-1.7B		
Base (Instruct)	49.8	34.5	25.7
SFT	47.3	36.1	22.9
GRPO	51.0	38.6	26.1
OPSD	51.6	40.0	28.1
mygray Sleep	53.2	40.2	29.3
	Qwen3-8B		
Base (Instruct)	73.8	68.1	42.4
SFT	75.5	66.4	43.7
GRPO	76.4	68.1	44.9
OPSD	76.6	67.4	45.1
mygray Sleep	79.2	69.0	46.1

E.3 Knowledge Incorporation

Table 4 evaluates the full Sleep design on integrating new factual information from SQuAD (Rajpurkar et al., 2016), following the setup of Zweiger et al. (2025). We compare against the base model, fine-tuned models without dreaming, and SEAL (Zweiger et al., 2025). Sleep achieves the best results in both the single-passage ($n = 1$) and continued pretraining ($n = 200$) settings. We also ablate individual components: removing dreaming causes the largest drop, confirming its importance for knowledge incorporation.

Table 4: Knowledge Incorporation Performance across Passage Settings

Method	**Single Passage** (n = 1)	**Continued Pretraining** (n = 200)
Base model	31.9	31.9
Fine-tuned Model with No Dreaming	33.4	32.0
SEAL	46.7	43.2
mygray Sleep (Transformer)	48.1	44.3
mygray Sleep (Transformer + four-level)	48.9	46.2
removing gradient-based selection	47.1	45.2
removing random expert	48.0	44.7
removing Dreaming	35.7	36.2
Dreaming only	47.4	43.6

E.4 Few-Shot Abstract Reasoning

Table 5 reports the few-shot abstract reasoning results following the ARC protocol from Akyürek et al. (2024a); Zweiger et al. (2025). Sleep achieves 80% success rate, substantially outperforming ICL (0%), TTT (10%), and SEAL (72.5%).

Table 5: Few-shot Abstract Reasoning

Method	**Success Rate (%)**
ICL	0
TTT	10
SEAL	72.5
mygray Sleep	80

F Sleep Terminology in the Literature

The human sleep process has been an inspiration for the design of many studies in the literature (McClelland et al., 1995; Kumaran et al., 2016; Hassabis et al., 2017; Ha & Schmidhuber, 2018b). In particular, "dreaming" has motivated several past studies in the literature to design methods that replay recent experiences/input data (Lin, 1992; Mnih et al., 2015; Ha & Schmidhuber, 2018a;b). Several studies designed an offline process that can make the model more robust for long-horizon tasks (Hafner et al., 2020; Tadros et al., 2022; González et al., 2020). Lin et al. (2025) suggest an offline self-study process as a form of "sleep-time compute" that summarizes the past context for the next user session, mainly is designed as a proposal for spending compute on the context rather than training the model itself.

To the best of our knowledge, the existing literature (including sleep-inspired studies) remains firmly anchored in the conventional distinction of training and testing phases. Even in the continual or online learning setups, models alternate between updating parameters (training) and evaluating performance (testing). We argue that for lifelong adaptation, the static train/test paradigm needs to be replaced with a continuous periodic "wake" and "sleep" lifecycle. During the "wake" phase, the model is actively interacting with varying input data and rapidly acquiring temporary information while during the "sleep" phase, the model consolidates its memory and internally processes existing knowledge.

G Implementation Note

Implementing the growing sparse modules can be extremely challenging if it requires a direct change in the dimensionality of tensors in the implementation. Alternatively, we can initially have those parameters in the model, but masked them in the forward and backward pass, before their initial activation in a sleep stage. Interestingly, it also aligns with our understanding of human brain, where brain has (large but) fixed capacity and new components are not added over time. Instead, new connections between brain regions can shape through our life, unlocking the activation of new neurons and resulting in more plasticity to learn new tasks (Kandell et al., 2021).