Language Models Need Sleep: Learning to Self-Modify and Consolidate Memories

Ali Behrouz [†], Farnoosh Hashemi [‡], Adel Javanmard [†], and Vahab Mirrokni [†]

[†]Google Research [‡]Cornell University

arXiv:2606.03979v2 [cs.LG] 10 Jul 2026

Abstract

The past few decades have witnessed significant advances in the design of machine learning algorithms–from early studies on task-specific shallow models to more general deep Large Language Models (LLMs). Despite showing promising results in tasks that require instant prediction or in-context learning, existing models lack the ability to continually learn and effectively transfer their temporal in-context knowledge to their long-term parameters. Inspired by human learning process, we introduce a "*Sleep*" paradigm that allows the models to continually learn, distill their short-term fragile memories into stable long-term knowledge with replay, and recursively improve themselves with "*Dreaming*" process. In more detail, sleep consists of two stages: (1) **Memory Consolidation**: an *upward* distillation process, called *Knowledge Seeding*, where the memories of a *smaller*-self are distilled into a *larger* network to provide more capacity while preserving the knowledge. As a proof of concept, we present a new *Generalized Distillation* process for Knowledge Seeding (i.e., the combination of on-policy distillation with Reinforcement Learning (RL)-based imitation learning); (2) **Dreaming**: a self-improvement phase, where the model uses RL to generate a curriculum of synthetic data to rehearse new knowledge and refine existing capabilities without human supervision. 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 (Nijkamp et al. 2023; Wang et al. 2023; Comanici et al. 2025), they are largely static after their initial deployment, meaning that they successfully perform tasks learned during pre- or post-training, but are unable to *continually acquire* new capabilities beyond their immediate context. This inherent static nature creates a crucial vulnerability: The model's knowledge and skills become progressively stale, operating with a fixed "knowledge cutoff" date beyond which it is unaware of new facts, events, and evolving information (Cheng et al. 2024).

Efforts to overcome this limitation have primarily focused on: (1) re-pretraining on an expanded dataset, which despite its effectiveness, is computationally expensive and impractical for frequent updates (Ibrahim et al. 2024); (2) using expensive continual parameter updates or other lightweight alternatives, such as fine-tuning or low-rank adaption (Hu et al. 2022; Akyürek et al. 2024a), which with iterative updates often results in Catastrophic Forgetting (CF) (Kemker et al. 2018; Shi et al. 2024)–a well-known phenomenon where the model's proficiency on original tasks degrades catastrophically as it learns new ones. This dilemma—between knowledge obsolescence on one hand and catastrophic forgetting as well as the prohibitive cost or destructive nature of updates on the other—underscores a critical, unresolved challenge: enabling LLMs to learn incrementally and efficiently throughout their lifecycle.

In recent years, In-Context Learning (ICL) (Brown et al. 2020) has gained attention as a highly efficient and successful form of continual learning (Akyürek et al. 2022, 2024b; Dong et al. 2024; Li et al. 2025). Initially, ICL was known as an emergent ability of LLMs that is trained on large scale data, enabling them to adapt fast to the context and so perform zero- or few-shot tasks (Brown et al. 2020). Later, more studies revealed and formalized the role of ICL as a meta-learning process in which the model performs internal computations along the sequence to incorporate context knowledge to its output by keeping or compress it into a short-term memory (Behrouz et al. 2025; Dherin et al. 2025). Despite the effectiveness/efficiency of ICL

[§]Correspondence to: {alibehrouz, adeljavanmard, mirrokni}@google.com and sh2574@cornell.edu.

[*]A version of this work has been publicly available from September 2025 on OpenReview.

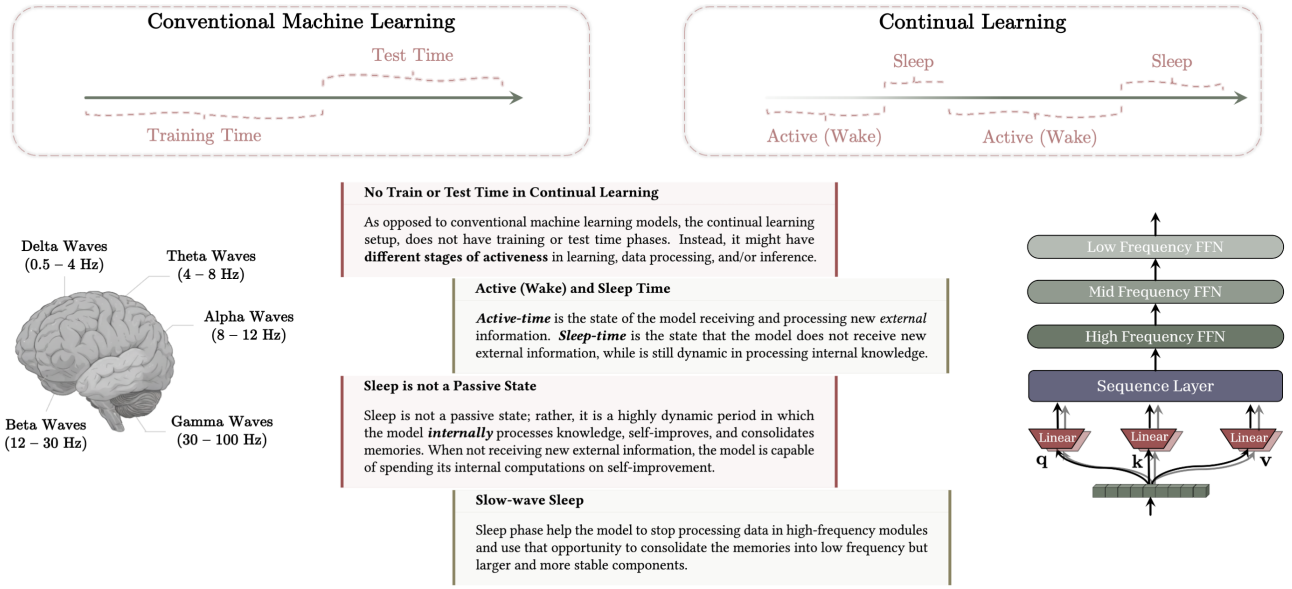

Figure 1: (**Conventional Machine Learning vs. Continual Learning**) While in conventional machine learning often the lifespan of the model is divided to *test and training time*, continual learning setup does not have these phases. We suggest that a continual learner need to have different stages of 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.

as a form of continual learning, it is limited to the context-window of sequence models, meaning that any new acquired knowledge will be removed from the model at the end of the session/context. This perspective raises a critical question: *How the model can effectively transfer the fragile short-term memories into more stable long-term knowledge*?

As an analogy, we use the example of anterograde amnesia for LLMs from Behrouz et al. (2025): Consider an impairment in the process of transferring the information from short-term to longer-term memories in humans, example of which is anterograde amnesia–a neurological condition where a person cannot form new memories after the onset of the disorder, while existing memories remain intact (Scoville et al. 1957). Such conditions can limit the person's knowledge to immediate present that fits in the short-term memory and long past, before the onset of the disorder, resulting in continuously experiencing the immediate present as if it were always new. One might notice a similar pattern in the memory processing of current Transformer-based LLMs. The knowledge of LLMs are limited in either: (1) the immediate context that fits into their context window (a.k.a. in-context learning), or (2) MLP and projection layers, storing long-past, before the onset of "end of pre-training." This similarity in pattern motivates us to ask, *What is the critical component in human learning process that consolidates memories?*

The human brain is highly efficient, lossy but effective when it comes to consolidating memory, which is often attributed to neuroplasticity—the brain's ability to change itself in response to new experiences, memories, and learning (Pascual-Leone et al. 2005; Johnston 2009). Recent studies support that long-term memory formation involves at least two distinct but complementary consolidation processes (Frey et al. 1997; Goto et al. 2021; Yang et al. 2024): (1) A rapid "online" consolidation phase occurs immediately or soon after learning, even during wakefulness. This is when new fragile memory traces are stabilized and begin transferring from short-term to more longer-term storage; (2) An "offline" consolidation (also known as systems consolidation) process repeats the replay of the recently encoded patterns during sleep and reorganizes the memory and supports transfer to cortical sites (Foster et al. 2006; Ji et al. 2007; Peyrache et al. 2009).

Online Consolidation in Humans (in Active Stage)

Online consolidation in humans is the active, wake-state strengthening and transformation of a newly learned memory by reactivating the memory during recall, making it more stable and more semantic-like over time.

Offline Consolidation in Humans (in Sleep and Quiet Rest Stages)

Offline consolidation in humans occurs after learning during rest or sleep, when a newly encoded memory is stabilized and gradually transformed into a more distributed neocortical representation by a complex lossy distillation process.

Returning to the analogy of anterograde amnesia, the evidence indicates that the condition can affect both stages. Behrouz et al. (2025), recently, presented the Nested Learning (NL) paradigm and aimed at the first form of memory consolidation (i.e., online consolidation). In particular, NL-based Hope architecture (see Figure 1) with the reactivation of the memory in a continuum memory system, where each component is updated with a different frequency but in an end-to-end manner, transfers the knowledge from fast unstable components to more stable low frequency modules in an online manner. Although this end-to-end process can directly transfer the knowledge to more stable components and so *postpone* forgetting, it still uses the same amount of model's capacity. This mainly happens due the fact that the model does keep the knowledge at the same level of abstraction without any additional lossy compression. Furthermore, this form of consolidation is not enough for robust continual learning: **(i)** Is selective and retrieval-dependent: it depends on active retrieval and so strengthens the parts of a memory that are repeatedly recalled; **(ii)** Depends on the context: the update caused by the online consolidation is based only on the context of the model, and so misses the higher-level understanding of new knowledge with existing one. In this paper, we focus on the second type of consolidation: i.e., offline consolidation via sleep, which is inspired by the sleep stage in humans:

The Role of Sleep in Human Learning Process

Sleep is not a passive state but a dynamic and highly structured period of brain activity essential for cognitive function (Rasch et al. 2013; Goldstein et al. 2014). During sleep, the brain orchestrates complex processes fundamental to learning, neural plasticity, self-improvement, and memory consolidation (Wamsley et al. 2011; Rasch et al. 2013; Goldstein et al. 2014). In humans, these processes are primarily governed by two critical and alternating stages of sleep: Rapid Eye Movement (REM) and Non-REM (NREM) sleep.

Non-Rapid Eye Movement Sleep (Slow-Wave Sleep): This stage, particularly its deepest phase known as slow-wave sleep, is characterized by synchronized, high-amplitude, low-frequency neural activity. Slow-wave sleep is associated with two primary functions crucial for learning: The first is synaptic homeostasis, a process that globally downscales synaptic strengths to counteract the net increase in connectivity from waking experiences, thereby maintaining metabolic balance and preventing neural saturation (Tononi et al. 2006).

The second core function is memory consolidation, the transformation of fragile, recent experiences into stable, long-term knowledge (Squire et al. 1995). This process is orchestrated through a sophisticated dialogue between the hippocampus and the neocortex (Squire et al. 2015). The hippocampus serves as a high-fidelity temporary storage system, capable of rapidly encoding specific daily experiences. In contrast, the neocortex is a vast, long-term repository better suited for the gradual learning of generalized rules and semantic knowledge from these experiences (Squire et al. 1995, 2015). During slow-wave sleep, the brain initiates a nightly dialogue between these structures that facilitates an intricate transfer of information. Notably, this transfer does not simply replay raw data; instead, it re-architects the knowledge acquired during waking hours, extracting abstractions and integrating them into a cohesive semantic network.

Rapid Eye Movement Sleep: Characterized by high-frequency, low-amplitude brain waves that resemble an awake state, REM sleep is most commonly associated with dreaming. Functionally, this stage is linked to the selective strengthening of newly formed synapses and the integration of new information with pre-existing emotional and semantic networks. Furthermore, it is hypothesized to play a role in simulating future scenarios to improve adaptive behavior.

In summary, the cyclical alternation between NREM and REM stages throughout the night is crucial. NREM sleep appears to consolidate and prune the day's experiences to build a more efficient knowledge base. Subsequently, REM sleep seems to operate on this refined base, exploring novel connections and strengthening salient neural pathways.

No Training or Test Time in Continual Learning

For a continual learner, there is no boarder and clear distinction between training and test time. The model only experiences two different states: when it receives information as input, or when it is an isolated learning system.

Active (Wake) and Sleep Phases in Continual Learning

In the active or wake time, continual learner receives new input data and process it as needed. In the sleep time, however, the model receives minimal (or none) input data and focus on processing existing knowledge to consolidate memories and self-improve.

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**: To mitigate catastrophic forgetting (CF) and to capture higher-levels of abstraction, in the memory consolidation phase, the model uses a periodic process in which it activates/unlocks new parameters and distills the knowledge from higher-frequency (i.e., faster updating) layers/modules to the newly unlocked parameters in more stable (lower-frequency) layers/modules. This process allows enough plasticity for new parameters while ensuring the stability of old parameters, preserving the old knowledge.

ii **Self-Improvement via Dreaming**: While the previous first stage of sleep ensures transferring the knowledge abstraction to longer-term memories, this stage is responsible for the process of recursive self-improvement. In particular, given the current state of the model, it generates a set of dreams (i.e., synthetically self-generated data) to improve its own performance with particular focus on the acquiring more proficiency on the 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) that allows each component to have its own frequency of update, we suggest a periodic and gradual parameter (de)activation process, where given a block and for each sleep step, we deactivate a set of parameters in a faster block (i.e., higher frequency block) and replace them with a set of newly activated parameters in the current block. This allows maintaining 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. 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 (e.g., some parameters are not active), distill the knowledge to a larger version of the model (e.g., where parameters are active). We then use SKS as a solution for memory consolidation, in which the model distills its knowledge from high-frequency layer/blocks to low-frequency blocks.
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. In particular, in our imitation learning process, we suggest that the teacher (i.e., a smaller version of model with (compressed) privilege information), generates synthetic data and then masks the sequence; Then, the student aims to predict the continuation of the sequence and gets its reward based on the distance of teacher generated data and its own prediction.
4. **Random Expert Selection for Dreaming**: The dreaming phase is responsible for understanding the relationship between different types of knowledge, and so we present a random low-rank expert selection in Mixture of Experts (MoE) (Wu et al. 2024) that aims to mix the knowledge by combining the knowledge of irrelevant experts for output computation.
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 effectiveness of Sleep paradigm as well as 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 $\theta = \{W_1^{(f_1)}, \ldots, W_{k_1}^{(f_1)}\} \cup \{W_1^{(f_2)}, \ldots, W_{k_2}^{(f_2)}\} \cup \ldots \{W_1^{(f_c)}, \ldots, W_{k_c}^{(f_c)}\}$, where parameter sets are sorted based on their weight update frequencies $f_1 \geq \ldots, \geq f_c$ (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, which at high-level indicates how often the parameters of a module are updated over time, CMS is formalized as a chain of MLP blocks $\texttt{MLP}^{(f_1)}(\cdot), \ldots, \texttt{MLP}^{(f_k)}(\cdot)$, each of which is associated with a chunk size of $C^{(\ell)} := \frac{\max_{\ell'} C^{(\ell')}}{f_\ell}$ 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, i.e., $\boldsymbol{\theta}^{(f_\ell)}$, are updated every $C^{(\ell)}$ steps: i.e., $\boldsymbol{\theta}_{i+1}^{(f_\ell)} = \boldsymbol{\theta}_i^{(f_\ell)} - \boldsymbol{e}_{i,\ell}$ 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 (e.g., $\nabla \mathcal{L}(\boldsymbol{\theta}_t^{(f_\ell)}; x_t)$ in gradient descent) in an end-to-end optimization of the neural network. The formulation of CMS is very broad and, depending on the task, the objective $\mathcal{L}(\cdot)$ can be changed (the same as the MLP blocks in Transformers). Due to this generality of 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 $C^{(\ell-1)}$. It is notable that Equation 2 provides an important interpretation: parameters $\boldsymbol{\theta}_t^{(f_\ell)}$ are responsible for compressing their own context into the their parameters and so they are a representative of abstract knowledge of their context (We refer to the Nested Learning paper (Behrouz et al. 2025) for more details).

In summary, in this perspective, the sequence model (e.g., attention (Vaswani et al. 2017) or other memory modules or RNNs (Katharopoulos et al. 2020; Behrouz et al. 2024)) acts as the short-term memory of the model since their high-frequency update can push the old knowledge to be forgotten, making space for new memories. On the other hand, CMS blocks act as a spectrum of memory modules, in which earlier blocks (higher-frequency) are shorter-term memories, while later blocks (and ultimately the last one with close to zero frequency) are longer-term memories. While actively updating this memory system can enhance the resistance to CF, the CF can happen when the update period of all models matched at some point (Behrouz et al. 2025). Therefore, it is crucial that before each update of a memory block, a mechanism consolidates the abstracted knowledge of that block to more stable parameters.

2.3 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 et al. 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 et al. 2018a,b). Several studies designed an offline process that can make the model more robust for long-horizon tasks (González et al. 2020; Hafner et al. 2020; Tadros et al. 2022). 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.

3 The Sleep Paradigm

3.1 Learning Phases in Continual Learning

As discussed earlier, a continual learner is always learning from the data/experience. Therefore, the conventional split of the machine learning models' lifecycle (i.e., test and train time) is not directly applicable in the continual learning setup. We suggest the use of "active or wake" time and "sleep" time as the two critical phases of a lifecycle of a continual learner. In particular, in the active or wake time, continual learner receives new input data and process it as needed. In the sleep time, however, the model receives minimal (or none) input data and focus on processing existing knowledge to consolidate memories and self-improve.

However, the process of memory consolidation is not limited to the sleep phase. In fact, following the discussion in Nested Learning (NL) (Behrouz et al. 2025), there are two forms of memory consolidations: (1) Online consolidation; and (2) Offline consolidation:

Online Consolidation in Neural Networks in Active (Wake) Stage

Online consolidation happens through the end-to-end learning process of neural learning modules (e.g., Hope architecture in Figure 1) in the wake or active phase, where earlier layers (faster blocks with higher frequency) transfer their knowledge to later layers (i.e., slower blocks with lower frequency).

Offline Consolidation in Neural Networks in Sleep Stage

In offline consolidation phase, the model enters the sleep phase, where it stops processing new data, but uses self-generated data to consolidate the recent memories.

In the next section, we present Sleep paradigm, in which contrary to the model's waking time (or active time), the model does not receive any external input data and concentrates its internal computations on self-improvement, consolidating

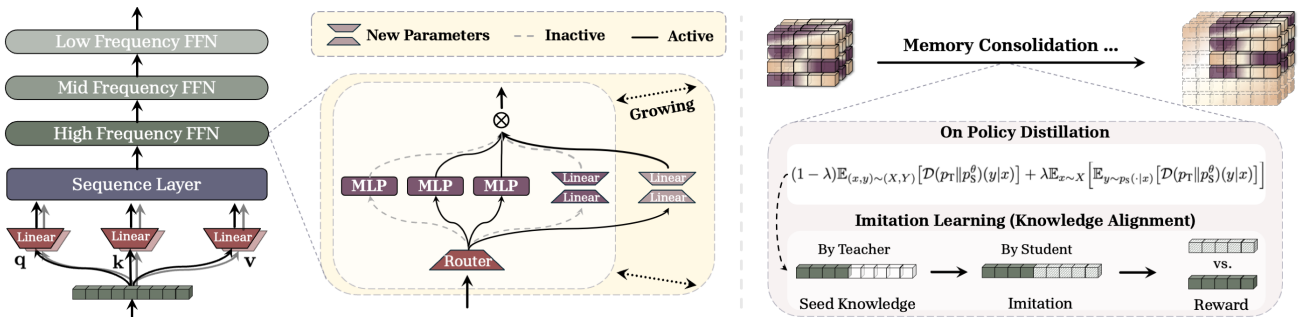

Figure 2: An overview of Memory Consolidation. The model increases its own number of parameters to enhance its capacity (Section 3.2), and then using the knowledge seeding, it transfers the knowledge abstractions from the higher to a lower-frequency memory (Section 3.3).

the past memories, and abstracting knowledge. In particular, we divide the sleep process into two key stages: (1) Memory consolidation; and (2) Dreaming for self-improvement.

3.2 Memory Consolidation: Parameter Expansion

As discussed earlier, in memory consolidation, we aim to transfer the short-term fragile memories into more vast and stable parameters. 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 $b \in \mathbb{N}$. Based on the update frequency of MLP blocks, we might need to consolidate the memory of a memory module into its next memory block multiple times. For example, consider a memory with update frequency of 1K followed by a memory with update frequency of 10K: in this case, the faster memory is updated 10 times before the update of slower memory block, which means 10 memory consolidation steps of faster memory to slower memory before slower memory's own update. This multiple consolidation steps into the slower memory can be a critical bottleneck to unlock continual learning capabilities of LLMs, due to the 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. The foundation of human's brain solution to this challenge is neuroplasticity, the brain's inherent ability to modify its own function and shape new connections in response to experiences. Inspired by this, 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. These new parameters will be allocated for storing the new transferred knowledge from $\texttt{MLP}^{(f_{\ell^*-1})}(\cdot)$. 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 $\texttt{MLP}^{(f_{\ell^*})}(\cdot)$. 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*) *U*pdate. 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.

> **Upward Distillation (Knowledge Seeding)**
>
> Given a set of small models $\mathcal{S}_1(\cdot), \ldots, \mathcal{S}_k(\cdot)$ as the teacher, Knowledge Seeding (KS) aims to transfer the knowledge in these models to a larger model $\mathcal{M}(\cdot)$.

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 et al. (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.

As discussed earlier, the memory consolidation step should not simply replay raw data; instead, it needs to explore and extract abstractions of knowledge acquired during active (waking) steps. To this end, knowledge seeding has two main steps: (1) A distillation process, in which student receives token-specific feedback from the teacher's logits on the self-generated sequences; and (2) An RL-based imitation learning method that forces the student to memorize the sampled outputs of teacher, aligning their sampling process while preserving the 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 do not backpropagate through the sampling distribution of the student, which can help with training stability and also speed. Also, 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:

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

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). For semantic similarity, we use a reward model that is frozen and rewards the student with 1 (resp. 0), if the semantic of $\hat{d}^{(i)}$ and $d^{(i)}$ are the same (resp. otherwise). On the other hand, absolute reward is defined based on the Levenshtein distance of the two sequences (denoted by $z(\cdot,\cdot)$): i.e., $r_{\text{abs}}(\hat{d}^{(i)}; d^{(i)}; \texttt{LM}_{\theta_{\text{exp}}})$ is defined as:

$$r_{\text{abs}}(\cdot) = \begin{cases} 1 - \frac{z(\hat{d}^{(i)}, d^{(i)})}{\max\{|\hat{d}^{(i)}|, |d^{(i)}|\}} & \text{if } z(\hat{d}^{(i)}, d^{(i)}) \leq z_0, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where z_0 is a similarity threshold. By incorporating the above LTI process to on-policy distillation, our on-policy knowledge seeding (KS) objective is defined as:

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

where $\alpha \in [0,1]$ controls the strength of the distillation compared to the LTI objective. Based on this objective, we update the new expanded parameters of the model and consolidate the memory/knowledge of high frequency memory into lower-frequency memory blocks. Now that the memories in $\texttt{MLP}^{(f_{t^*-1})}(\cdot)$ are consolidated in $\texttt{MLP}^{(f_{t^*})}(\cdot)$, we reset all the low-rank parameters that previously (in past sleep periods) have been added to $\texttt{MLP}^{(f_{t^*-1})}(\cdot)$, making its capacity available for future. This step, can be interpreted as a similar procedure of synaptic pruning in human brain, in which brain prunes connections that are unnecessarily and/or redundant (Li et al. 2017) to enhance its efficiency and performance.

Note on the Implementation. 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).

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.

In practice, any synthetic data generation process for self-improvement (e.g., Pang et al. (2024a), Huang et al. (2025), and 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

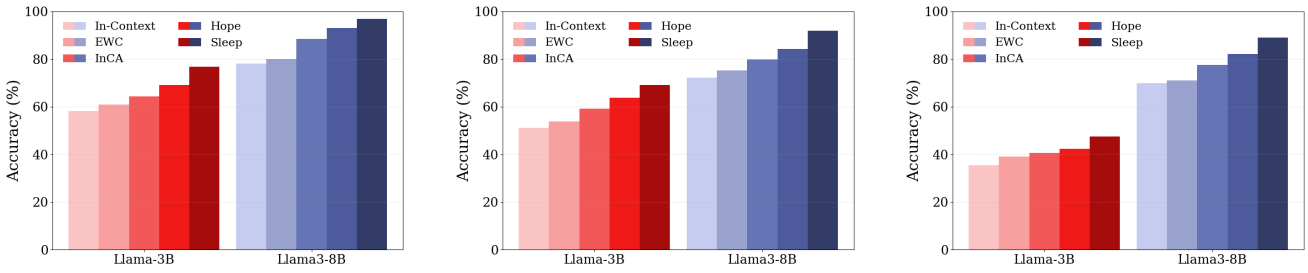

Figure 3: Class-incremental learning for text classification is evaluated on the (Left) CLINC dataset (Larson et al. 2019), (Middle) Banking dataset (Casanueva et al. 2020), and (Right) DBpedia dataset (Auer et al. 2007). The HOPE architecture consistently outperforms other continual learning approaches, achieving the highest accuracy.

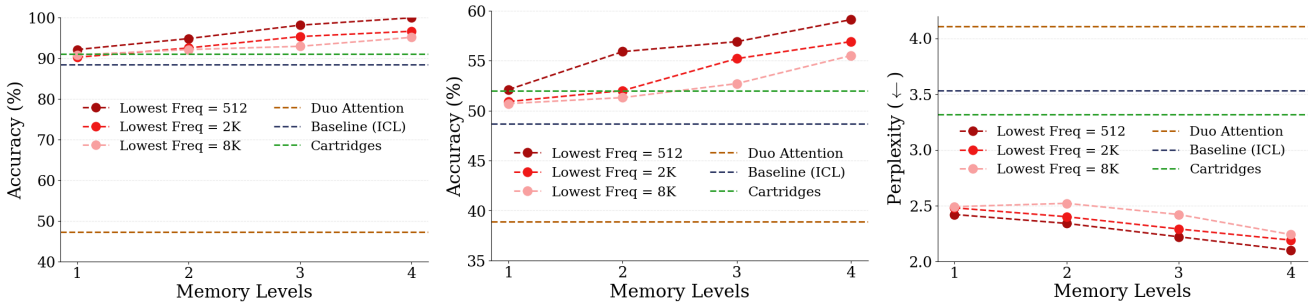

Figure 4: Effect of memory levels on in-context learning performance for (Left) MK-NIAH from RULER (Hsieh et al. 2024), (Middle) LongHealth (Adams et al. 2025), and (Right) QASPER (Dasigi et al. 2021). Lower values indicate better performance for QASPER.

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 (Pan et al. 2024; Wang et al. 2024a): for each dream, $\texttt{DREAM}^{(i)}$, we assign an importance score $\boldsymbol{\omega}^{(i)}$ and select 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 Empirical Results

In our empirical evaluation, we study the effect of each stage of the sleep and also all the stages together. In the first section, we focus on the memory consolidation, which is designed with the goal of enhancing the continual learning abilitites and long-context understanding. See Appendix B for additional experimental results.

4.1 The Effect of Memory Consolidation

One of the main goals of memory consolidation process is to improve continual learning while strengthening long-context understanding. In this section, we evaluate *only* memory consolidation phase (without self-improvement) on continual-learning and long-context benchmarks.

Class Incremental Learning. We first focus on class-incremental learning on three datasets of CLINC (Larson et al. 2019), Banking (Casanueva et al. 2020), and DBpedia (Auer et al. 2007) (see Section B.1 for the details). We use Llama-3B and Llama3-8B (Dubey et al. 2024) as the backbones. Hope is augmented with our memory consolidation mechanism, which improves abstraction via on-policy self-distillation. Following Momeni et al. (2025) we compare against ICL (same continual pre-training process but without sleep), Elastic Weight Consolidation (EWC) (Kirkpatrick et al. 2017), and In-context Continual Learning with an External Learner (InCA) (Momeni et al. 2025). We also include Hope (Behrouz et al. 2025) as a multi-level in-context updating baseline without explicit distillation process. Results in Figure 3 show that Hope performs best across datasets, including over external-learner (InCA) and regularization (EWC) approaches. Relative to ICL, gains come from converting prompt-level adaptation into durable parametric memory through consolidation. Relative to Hope, explicit self-distillation yields better abstractions than repeated in-context updates alone.

The Effect of Levels (#Sleep Phases) 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 LongHealth (Adams et al. 2025), QASPER (Dasigi et al. 2021), and MK-NIAH (Hsieh et al. 2024) datasets (see Section B.1). As baselines, we use ICL, DuoAttention (Xiao et al. 2025), and Cartridges (Eyuboglu et al. 2025). For Hope variants, we vary the sleep schedule by changing (i) how many consolidation stages are used and (ii) the persistence of the most stable memory, operationalized by the lowest consolidation frequency. Intuitively, a lower frequency yields more persistent but less adaptive long-term memory. Results are reported in Figure 4. Across all tasks, Hope consistently outperforms ICL and the efficient DuoAttention baseline, showing that sleep-time consolidation improves long-context behavior beyond prompt-only adaptation. We also observe that Hope outperforms Cartridges (Eyuboglu et al. 2025): while Cartridges improves efficiency by using an auxiliary model to compress KV representations, Hope instead performs on-policy self-distillation during sleep, consolidating newly acquired information into transferable parametric knowledge and yielding more robust long-context understanding. Comparing Hope variants with each other, we find two consistent trends: (1) increasing the number of consolidation stages improves in-context learning and long-context understanding, supporting the view that sleep enables better knowledge abstraction and compression, allowing the model to retain more information with fewer effective parameters; and (2) increasing the lowest frequency reduces performance, suggesting that making the most persistent memory more adaptive weakens retention.

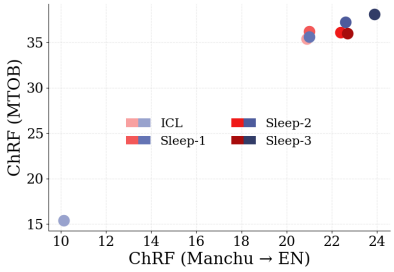

Figure 5: Continual Translation of a Novel Language (CTNL) task. Red points show performance when training on a single language, whereas blue points show performance under continual learning.

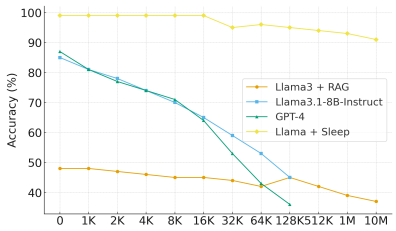

Figure 6: Results on the BABILong benchmark. Red points correspond to fine-tuned models, whereas blue points correspond to zero-shot evaluations of large-scale models.

Table 1: 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
Qwen3-8B			
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
OPSD	76.6	67.4	45.1
+ Expansion	77.9	68.2	45.9

Learning a New Language In-Context. LLMs often fail in continual settings, where models are expected to acquire new skills sequentially without overwriting previously acquired knowledge. To study this challenge, we follow the task introduced by Behrouz et al. (2025), which combines MTOB (Tanzer et al. 2024) and Manchu (Pei et al. 2025) datasets, two translation datasets for unseen languages during pre-training. That is, models are exposed in-context to two previously

unseen languages and must translate phrases into English. We consider two setups: learning and evaluating each language independently, and sequentially learning both languages before evaluating translation performance on each. We compare standard ICL with variants of Hope that differ in the number of sleep-time consolidation stages, denoted as Hope-1, Hope-2, and Hope-3. Results are shown in Figure 5, where each point reports ChRF scores for Manchu→English (x-axis) and Kalamang→English (y-axis). In the single-language setting, all Hope variants match or exceed ICL, indicating that consolidation does not hinder in-context adaptation. Under continual learning setup, ICL exhibits a sharp performance drop, largely reverting to pre-trained behavior, whereas Hope retains substantially more of its gains. Performance improves monotonically with additional consolidation stages, and Hope-3 nearly recovers its single-language performance despite sequential exposure. These results underscore the role of sleep-time consolidation in continual learning. Unlike ICL's prompt-level updates, Sleep introduces an explicit self-improvement phase that distills useful abstractions into longer-lasting parametric memory, enabling effective sequential learning without catastrophic forgetting. As another baseline, we also evaluated Cartridges (Eyuboglu et al. 2025) and Supervised Fine-Tuning (SFT) over the languages. Surprisingly, both methods faced catastrophic forgetting in at least on of the languages, performing even weaker than ICL in at least one of the tasks (placing them outside of the plot in Figure 5).

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). Detailed discussion on the results can be found in Section B.2. In summary, Hope achieve almost perfect score in scaling to 10M of tokens.

Memory Consolidation for Reasoning. Another important implication of memory consolidation phase is improving the reasoning capability of the model. In this section, we evaluate its effect of mathematical reasoning and compare it with common baselines of base model, SFT, and GRPO (Shao et al. 2024). The results are reported in Table 2. Our algorithm for memory consolidation performs better than SFT and GRPO for improving the reasoning capabilities of the base model.

Table 2: 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
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
Sleep	79.2	69.0	46.1

Table 3: 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
Sleep (Transformer)	48.1	44.3
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

Table 4: Few-shot Abstract Reasoning

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

Knowledge Incorporation. Next, we focus on the full design of Sleep, allowing for self-improvement as well. In this task, we expect the model to be able to answer questions about the incorporated facts. We follow the experimental setup of Zweiger et al. (2025), including the choice of models and parameters, for the sake of fair comparison. We evaluate our model on integrating new factual information from SQuAD dataset (Rajpurkar et al. 2016). As baselines, we use (i) a base model, which is the variant without any improvement or having access to the passage; (ii) a fine-tuned model with no dreaming, (iii) SEAL model with RL and self-adaption; (iv) our Transformer-based architecture with two level memory system; and (v) with four-level memory system.

Table 3 summarizes mean no-context SQuAD accuracy for both the single-passage ($n = 1$) and continued pretraining (CPT, $n = 200$) settings. Our sleep process achieves the best results among other settings and state-of-the-art methods like SEAL. We attributes this results to: (1) memory consolidation steps that let the model store its knowledge more effectively; (2) our improvements on top of the SEAL that we discussed in Section 3.4. In the CPT regime, the model is exposed to $n = 200$ passages during a single continued pretraining run and is evaluated on the full set of 974 associated questions. For each passage, we sample five dreams and combine them into an aggregated synthetic dataset for training. Sleep process achieves the best performance.

Few-Shot Learning. 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. See Section B.3 for the details. In this setting (see Table 4), Sleep achieves a 80% success rate, higher then the other methods.

4.2 Ablations on the Design Choices

We ablate the design choices for the memory consolidation process on the mathematical reasoning. The results are reported in Figure 1. All the components contribute positively to the performance of our method. We also, further ablate the design choices on self-improvement in Table 3.

5 Conclusion

In this work, we introduced the Sleep paradigm for Large Language Models, consists of: (i) *knowledge seeding*, an upward distillation that transfers short-term, in-context knowledge into lower-frequency, long-term parameters, and (ii) *dreaming*, self-generated training that improves capabilities while controlling interference. In our experimental results, across long-context understanding, knowledge incorporation, few-shot reasoning, and continual learning, Sleep yields consistent gains.

References

[1] William Beecher Scoville and Brenda Milner. "Loss of recent memory after bilateral hippocampal lesions". In: *Journal of neurology, neurosurgery, and psychiatry* 20.1 (1957), p. 11.

[2] Jürgen Schmidhuber. *Evolutionary Principles in Self-Referential Learning*. 1987. URL: https://people.idsia.ch/~juergen/diploma1987ocr.pdf.

[3] Long-Ji Lin. "Self-improving reactive agents based on reinforcement learning, planning and teaching". In: *Machine Learning* 8.3–4 (1992), pp. 293–321.

[4] Juergen Schmidhuber. "Learning to control fast-weight memories: An alternative to recurrent nets. Accepted for publication in". In: *Neural Computation* (1992).

[5] 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". In: *Psychological Review* 102.3 (1995), pp. 419–457.

[6] Larry R Squire and Pablo Alvarez. "Retrograde amnesia and memory consolidation: a neurobiological perspective". In: *Current opinion in neurobiology* 5.2 (1995), pp. 169–177.

[7] Uwe Frey and Richard GM Morris. "Synaptic tagging and long-term potentiation". In: *Nature* 385.6616 (1997), pp. 533–536.

[8] Alvaro Pascual-Leone, Amir Amedi, Felipe Fregni, and Lotfi B Merabet. "The plastic human brain cortex". In: *Annu. Rev. Neurosci.* 28.1 (2005), pp. 377–401.

[9] Robert Stickgold. "Sleep-dependent memory consolidation". In: *Nature* 437.7063 (2005), pp. 1272–1278.

[10] David J Foster and Matthew A Wilson. "Reverse replay of behavioural sequences in hippocampal place cells during the awake state". In: *Nature* 440.7084 (2006), pp. 680–683.

[11] Giulio Tononi and Chiara Cirelli. "Sleep function and synaptic homeostasis". In: *Sleep medicine reviews* 10.1 (2006), pp. 49–62.

[12] 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*. Springer. 2007, pp. 722–735.

[13] Daoyun Ji and Matthew A Wilson. "Coordinated memory replay in the visual cortex and hippocampus during sleep". In: *Nature neuroscience* 10.1 (2007), pp. 100–107.

[14] Michael V Johnston. "Plasticity in the developing brain: implications for rehabilitation". In: *Developmental disabilities research reviews* 15.2 (2009), pp. 94–101.

[15] 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". In: *Nature neuroscience* 12.7 (2009), pp. 919–926.

[16] Erin J Wamsley and Robert Stickgold. "Memory, sleep and dreaming: experiencing consolidation". In: *Sleep medicine clinics* 6.1 (2011), p. 97.

[17] Björn Rasch and Jan Born. "About sleep's role in memory". In: *Physiological reviews* (2013).

[18] Andrea N Goldstein and Matthew P Walker. "The role of sleep in emotional brain function". In: *Annual review of clinical psychology* 10.1 (2014), pp. 679–708.

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531* (2015).

[20] 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". In: *Nature* 518.7540 (2015), pp. 529–533.

[21] Larry R Squire, Lisa Genzel, John T Wixted, and Richard G Morris. "Memory consolidation". In: *Cold Spring Harbor perspectives in biology* 7.8 (2015), a021766.

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

[23] Dharshan Kumaran, Demis Hassabis, and James L. McClelland. "What learning systems do intelligent agents need? Complementary learning systems theory updated". In: *Trends in Cognitive Sciences* 20.7 (2016), pp. 512–534.

[24] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Ed. by Jian Su, Kevin Duh, and Xavier Carreras. Association for Computational Linguistics, 2016. URL: https://aclanthology.org/D16-1264/.

[25] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Proceedings of Machine Learning Research. PMLR, 2017. URL: https://proceedings.mlr.press/v70/finn17a.html.

[26] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. "Neuroscience-inspired artificial intelligence". In: *Neuron* 95.2 (2017), pp. 245–258.

[27] 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". In: *Proceedings of the national academy of sciences* 114.13 (2017), pp. 3521–3526.

[28] Wei Li, Lei Ma, Guang Yang, and Wen-Biao Gan. "REM sleep selectively prunes and maintains new synapses in development and learning". In: *Nature neuroscience* 20.3 (2017), pp. 427–437.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[30] David Ha and Jürgen Schmidhuber. "Recurrent world models facilitate policy evolution". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 31. 2018, pp. 2451–2463.

[31] David Ha and Jürgen Schmidhuber. "World models". In: *arXiv preprint arXiv:1803.10122* 2.3 (2018), p. 440.

[32] 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*. Vol. 32. 1. 2018.

[33] 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)*. 2019, pp. 1311–1316.

[34] Noam Shazeer. "Fast transformer decoding: One write-head is all you need". In: *arXiv preprint arXiv:1911.02150* (2019).

[35] 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". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[36] Inigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. "Efficient Intent Detection with Dual Sentence Encoders". In: *ACL 2020* (2020), p. 38.

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

[38] 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.

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

[40] 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". In: *arXiv preprint arXiv:2105.03011* (2021).

[41] 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". In: *Science* 374.6569 (2021), pp. 857–863. DOI: 10.1126/science.abj9195. eprint: https://www.science.org/doi/pdf/10.1126/science.abj9195. URL: https://www.science.org/doi/abs/10.1126/science.abj9195.

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

[43] Brian Lester, Rami Al-Rfou, and Noah Constant. "The power of scale for parameter-efficient prompt tuning". In: *arXiv preprint arXiv:2104.08691* (2021).

[44] Xiang Lisa Li and Percy Liang. "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *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)*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto

Navigli. Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597. DOI: 10.18653/v1/2021.acl-long.353. URL: https://aclanthology.org/2021.acl-long.353/.

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

[46] Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. "Recurrent memory transformer". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 11079–11091.

[47] 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.

[48] 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.

[49] Timothy Tadros, Giri P. Krishnan, Ramyaa Ramyaa, and Maxim Bazhenov. "Sleep-like unsupervised replay reduces catastrophic forgetting in artificial neural networks". In: *Nature Communications* 13.1 (2022), p. 7742.

[50] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. "STaR: Bootstrapping Reasoning With Reasoning". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Curran Associates, Inc., 2022. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/639a9a172c044fbb64175b5fad42e9a5-Paper-Conference.pdf.

[51] 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". In: *arXiv preprint arXiv:2303.08774* (2023).

[52] 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". In: *arXiv preprint arXiv:2305.13245* (2023).

[53] Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. "Adapting language models to compress contexts". In: *arXiv preprint arXiv:2305.14788* (2023).

[54] 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". In: *arXiv preprint arXiv:2307.06945* (2023).

[55] 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". In: *arXiv preprint arXiv:2312.12379* (2023).

[56] Albert Gu and Tri Dao. "Mamba: Linear-time sequence modeling with selective state spaces". In: *arXiv preprint arXiv:2312.00752* (2023).

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

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

[59] Yucheng Li. "Unlocking context constraints of llms: Enhancing context efficiency of llms with self-information-based content filtering". In: *arXiv preprint arXiv:2304.12102* (2023).

[60] 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_.

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

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

[63] 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". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 61501–61513.

[64] 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". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 34661–34710.

[65] 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". In: *arXiv preprint arXiv:2412.08905* (2024).

[66] 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.

[67] 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*. 2024.

[68] Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. "In-context language learning: Architectures and algorithms". In: *arXiv preprint arXiv:2401.12973* (2024).

[69] 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.

[70] 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". In: *arXiv preprint arXiv:2405.04517* (2024).

[71] Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. "Titans: Learning to memorize at test time". In: *arXiv preprint arXiv:2501.00663* (2024).

[72] 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.

[73] 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: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 1107–1128. DOI: 10.18653/v1/2024.emnlp-main.64. URL: https://aclanthology.org/2024.emnlp-main.64/.

[74] 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". In: *arXiv e-prints* (2024), arXiv–2407.

[75] 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.

[76] 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". In: *Transactions on Machine Learning Research* (2024). ISSN: 2835-8856. URL: https://openreview.net/forum?id=DimPeeCxKO.

[77] Kalle Kujanpää, Harri Valpola, and Alexander Ilin. "Knowledge injection via prompt distillation". In: *arXiv preprint arXiv:2412.14964* (2024).

[78] 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". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 106519–106554.

[79] 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". In: *arXiv preprint arXiv:2404.15159* (2024).

[80] 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". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 22947–22970.

[81] Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. "MiniCache: KV Cache Compression in Depth Dimension for Large Language Models". In: *Advances in Neural Information Processing Systems* 37 (2024).

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

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

[84] 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: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 15395–15406. DOI: 10.18653/v1/2024.acl-long.821. URL: https://aclanthology.org/2024.acl-long.821/.

[85] 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*. 2024. URL: https://openreview.net/forum?id=38E4yUbrgr.

[86] 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*. 2024. URL: https://openreview.net/forum?id=38E4yUbrgr.

[87] Ivan Rodkin, Yuri Kuratov, Aydar Bulatov, and Mikhail Burtsev. "Associative recurrent memory transformer". In: *arXiv preprint arXiv:2407.04841* (2024).

[88] 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". In: *arXiv preprint arXiv:2402.03300* (2024).

[89] 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". In: *ACM Computing Surveys* (2024).

[90] 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". In: *Transactions on Machine Learning Research* (2024). Expert Certification. ISSN: 2835-8856. URL: https://openreview.net/forum?id=lNAyUngGFK.

[91] 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". In: *Transactions on Machine Learning Research* (2024). URL: https://openreview.net/forum?id=lNAyUngGFK.

[92] 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". In: *arXiv preprint arXiv:2407.04620* (2024).

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

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

[95] 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.

[96] 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". In: *arXiv preprint arXiv:2406.13035* (2024).

[97] 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". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 131197–131223.

[98] 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". In: *arXiv preprint arXiv:2407.08454* (2024).
[99] Xun Wu, Shaohan Huang, and Furu Wei. "Mixture of lora experts". In: *arXiv preprint arXiv:2404.13628* (2024).
[100] 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". In: *arXiv preprint arXiv:2409.02877* (2024).
[101] 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". In: *arXiv preprint arXiv:2408.07057* (2024).
[102] Wannan Yang, Chen Sun, Roman Huszár, Thomas Hainmueller, Kirill Kiselev, and György Buzsáki. "Selection of experience for memory by hippocampal sharp wave ripples". In: *Science* 383.6690 (2024), pp. 1478–1483.
[103] 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". In: *arXiv preprint arXiv:2410.03111* (2024).
[104] 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*. 2024.
[105] 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". In: *arXiv preprint arXiv:2402.09997* (2024).
[106] 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". In: *arXiv preprint arXiv:2409.16167* (2024).
[107] 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". In: *Journal of Healthcare Informatics Research* (2025), pp. 1–17.
[108] 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.
[109] Lucas Caccia, Alan Ansell, Edoardo Ponti, Ivan Vulic, and Alessandro Sordoni. "Training Plug-n-Play Knowledge Modules with Deep Context Distillation". In: *arXiv preprint arXiv:2503.08727* (2025).
[110] Vivek Chari, Guanghui Qin, and Benjamin Van Durme. "KV-Distill: Nearly Lossless Learnable Context Compression for LLMs". In: *arXiv preprint arXiv:2503.10337* (2025).
[111] 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". In: *arXiv preprint arXiv:2507.06261* (2025).
[112] DeepSeek-AI. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. arXiv: 2501.12948 [cs.CL]. URL: https://arxiv.org/abs/2501.12948.
[113] Benoit Dherin, Michael Munn, Hanna Mazzawi, Michael Wunder, and Javier Gonzalvo. "Learning without training: The implicit dynamics of in-context learning". In: *arXiv preprint arXiv:2507.16003* (2025).
[114] 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". In: *arXiv preprint arXiv:2506.06266* (2025).
[115] 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.
[116] Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. "Long-context LLMs Struggle with Long In-context Learning". In: *Transactions on Machine Learning Research* (2025). ISSN: 2835-8856. URL: https://openreview.net/forum?id=Cw2xlg0e46.
[117] 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". In: *arXiv preprint arXiv:2504.13171* (2025).
[118] 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". In: *arXiv preprint arXiv:2502.14644* (2025).

[119] 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*. 2025, pp. 7292–7306.

[120] 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". In: *arXiv preprint arXiv:2502.11862* (2025).

[121] Haris Riaz, Sourav Bhabesh, Vinayak Arannil, Miguel Ballesteros, and Graham Horwood. "MetaSynth: Meta-Prompting-Driven Agentic Scaffolds for Diverse Synthetic Data Generation". In: *arXiv preprint arXiv:2504.12563* (2025).

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

[123] 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.

[124] 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.

[125] Adam Zweiger, Jyothish Pari, Han Guo, Ekin Akyürek, Yoon Kim, and Pulkit Agrawal. "Self-Adapting Language Models". In: *arXiv preprint arXiv:2506.10943* (2025).

[126] 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. arXiv: 2604.12002 [cs.CL]. URL: https://arxiv.org/abs/2604.12002.

[127] 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". In: *arXiv preprint arXiv:2601.20802* (2026).

[128] 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. arXiv: 2603.24472 [cs.CL]. URL: https://arxiv.org/abs/2603.24472.

[129] John Kirchenbauer, Abhimanyu Hans, Brian Bartoldson, Micah Goldblum, Ashwinee Panda, and Tom Goldstein. *Multi-Token Prediction via Self-Distillation*. 2026. arXiv: 2602.06019 [cs.CL]. URL: https://arxiv.org/abs/2602.06019.

[130] Yihong Liu, Raoyuan Zhao, Michael A. Hedderich, and Hinrich Schütze. *Crosslingual On-Policy Self-Distillation for Multilingual Reasoning*. 2026. arXiv: 2605.09548 [cs.CL]. URL: https://arxiv.org/abs/2605.09548.

[131] Ruiyang Qin, Qingzhuo Wang, Dongrui Liu, Qiang Li, Zhihua Wei, and Wen Shen. *Multilingual Safety Alignment via Self-Distillation*. 2026. arXiv: 2605.02971 [cs.LG]. URL: https://arxiv.org/abs/2605.02971.

[132] Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. *CRISP: Compressed Reasoning via Iterative Self-Policy Distillation*. 2026. arXiv: 2603.05433 [cs.LG]. URL: https://arxiv.org/abs/2603.05433.

[133] Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. "Self-Distillation Enables Continual Learning". In: *arXiv preprint arXiv:2601.19897* (2026).

[134] Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. *Self-Distillation Enables Continual Learning*. 2026. arXiv: 2601.19897 [cs.LG]. URL: https://arxiv.org/abs/2601.19897.

[135] Alex Stein, Furong Huang, and Tom Goldstein. *GATES: Self-Distillation under Privileged Context with Consensus Gating*. 2026. arXiv: 2602.20574 [cs.LG]. URL: https://arxiv.org/abs/2602.20574.

[136] 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. arXiv: 2604.10674 [cs.LG]. URL: https://arxiv.org/abs/2604.10674.

[137] Tianzhu Ye, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. *Online Experiential Learning for Language Models*. 2026. arXiv: 2603.16856 [cs.CL]. URL: https://arxiv.org/abs/2603.16856.

[138] Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. *On-Policy Context Distillation for Language Models*. 2026. arXiv: 2602.12275 [cs.CL]. URL: https://arxiv.org/abs/2602.12275.

[139] Ruixiang Zhang, Richard He Bai, Huangjie Zheng, Navdeep Jaitly, Ronan Collobert, and Yizhe Zhang. *Embarrassingly Simple Self-Distillation Improves Code Generation*. 2026. arXiv: 2604.01193 [cs.CL]. URL: https://arxiv.org/abs/2604.01193.

[140] 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". In: *arXiv preprint arXiv:2604.17535* (2026).

[141] 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*. 2026. arXiv: 2604.17535 [cs.CL]. URL: https://arxiv.org/abs/2604.17535.

[142] 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*. 2026. arXiv: 2604.14054 [cs.LG]. URL: https://arxiv.org/abs/2604.14054.

[143] 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". In: *arXiv preprint arXiv:2601.18734* (2026).

[144] 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*. 2026. arXiv: 2601.18734 [cs.LG]. URL: https://arxiv.org/abs/2601.18734.

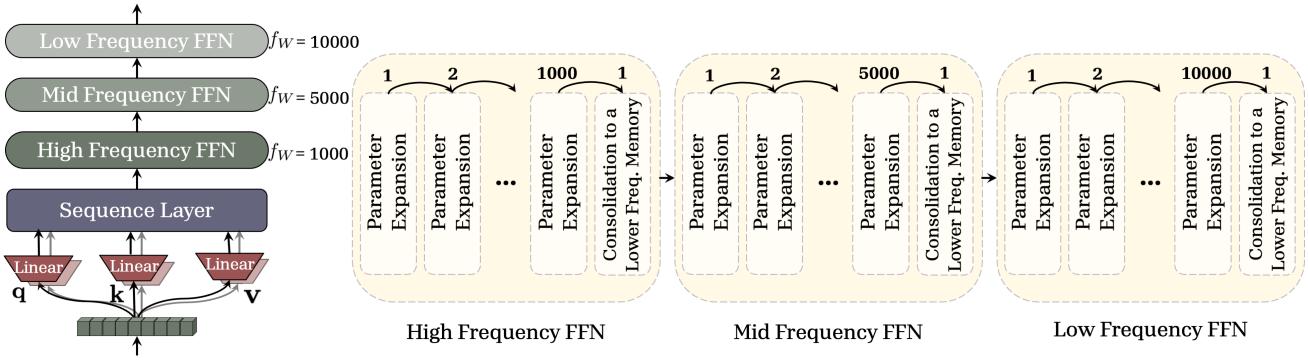

Figure 7: Multi-frequency memory hierarchy. Updates enter the High-Frequency FFN via repeated Parameter Expansion; when the window f_W expires, knowledge is Consolidated to the Mid- and then Low-Frequency FFNs (1k→5k→10k).

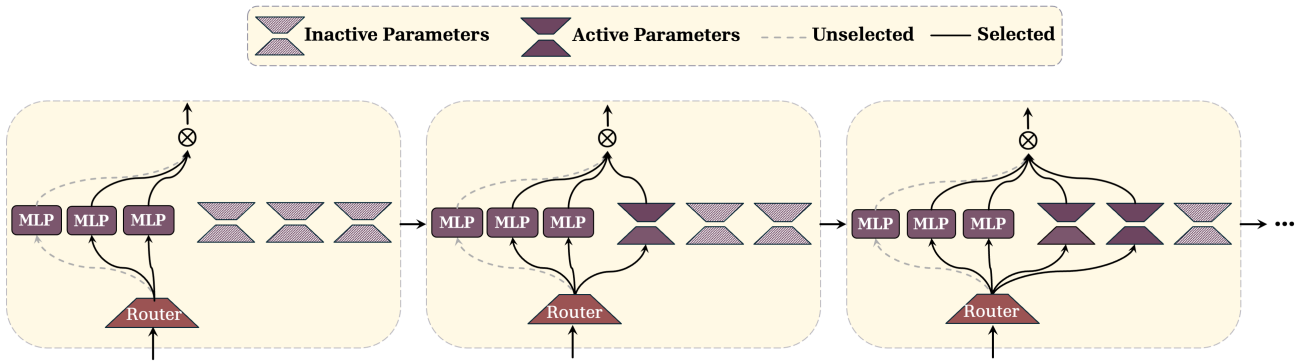

Figure 8: Memory consolidation by routed expert updates. Across Sleep cycles (left→right), a router selects and updates a small set of experts (solid), leaving others inactive (hatched), expanding capacity while limiting interference.

A Related Work

A.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 (Lester et al. 2021; Li 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 (Gou et al. 2023; Li et al. 2024a; Wu et al. 2024; Xiao et al. 2024; Yadav et al. 2024; Zhao et al. 2024b).

A.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; Caccia et al. 2025; Mao 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 (Zelikman et al. 2022; Singh et al. 2024b; DeepSeek-AI 2025) and self-rewarding mechanisms (Pang et al. 2024b; Wang et al. 2025). 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 (Abdin et al. 2024; Nayak et al. 2024; Riaz et al. 2025).

A recent group of concurrent studies has suggested to use on-policy distillation for self-distillation process (Hübotter et al. 2026; Zhang et al. 2026b; Zhao et al. 2026a) 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.

A.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) (Jiang et al. 2023; Li 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 (Zhang et al. 2023; Li et al. 2024b; Tang et al. 2024) or merging tokens based on similarity and attention density (Liu et al. 2024; Wan et al. 2024; Wang et al. 2024b; Zhang 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 (Gu et al. 2023; Arora et al. 2024; 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.

A.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× 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 (He et al. 2026; Wang 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 (Kim et al. 2026; Zhao et al. 2026b), demonstration (Shenfeld et al. 2026b), document (Stein et al. 2026), context or system prompt (Ye et al. 2026a,b), "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 (Liu et al. 2026; Qin 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 (He et al. 2026; Kim et al. 2026). **(iv) Knowledge seeding augments on-policy distillation with imitation learning.** Where the on-policy distillation objective in works such as (Liu et al. 2026; Qin et al. 2026; Sang et al. 2026; Stein et al. 2026; Ye et al. 2026a,b; Zhang et al. 2026c; Zhao et al. 2026b) 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.

B 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).

B.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.

B.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 6, 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.

B.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.

B.4 Additional Details

Table 5: 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

B.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.