TOWARDS CONTINUAL IN-CONTEXT LEARNING VIA TRAINING CARTRIDGES AT TEST TIME

Anonymous authors
Paper under double-blind review

ABSTRACT

Language model agents can learn from experience through in-context learning (ICL), but ICL is limited by finite context windows and declining performance as context grows. We formulate *continual in-context learning* as preserving ICL's performance and adaptability as experience grows beyond these limits. Our controlled experiments identify two requirements for iterative key–value (KV) cache compaction: each memory must preserve knowledge as new context arrives, and memories from successive compactions must work together. To meet these requirements, we introduce ACCORDION, which repeatedly consolidates an agent's oldest experiences into trained KV memory modules while keeping the base model frozen. The agent revisits these experiences to generate self-study examples, then distills them into a compact cache initialized with improved Attention Matching. Each new module is trained alongside frozen earlier modules and varying amounts of subsequent context, preparing it to remain useful as experience accumulates. Across QuALITY, Blind Spectrum Monitoring from Continual Learning Bench, and Harvey LAB, ACCORDION achieves relative improvements of approximately 13%, 60%, and 18%, respectively, over the strongest evaluated baselines using the same base model. It supports repeated compaction at 19–23$\times$ compression and handles tasks over corpora of 1M–12M tokens through compact memory and tool use. These results show that agents can preserve useful knowledge beyond their context windows and build on it to improve task performance.

1 INTRODUCTION

Language model agents learn from experience through in-context learning (ICL), adapting their behavior as new observations and feedback enter the context (Brown et al., 2020). This allows an agent to learn as it acts, but the history supporting that learning grows with every interaction. Over long trajectories, this history exceeds the model's context window, and performance can degrade even before the window fills through context rot (Hong et al., 2025). We formulate **continual in-context learning** as the problem of preserving ICL's performance and adaptability as an agent's experience grows beyond its context window.

A common way to keep an agent running is to summarize its history into text and continue from the summary (Wang et al., 2025a; Packer et al., 2023). In our evaluations, however, these summaries lose much of the original detail at high compression ratios. An alternative is to encode that information directly in continuous key–value (KV) vectors, building on prefix tuning (Li & Liang, 2021). The model can attend to these vectors without changes to its pretrained weights, and the retained information need not be expressed as a short text summary. Cartridges uses this representation to compress context through self-distillation on self-generated question–answer pairs (Eyuboglu et al., 2026), matching ICL with 38.6$\times$ less memory than the full context. Attention Matching instead constructs a compact cache directly by matching the original attention mass and output (Zweiger et al., 2026).

These methods suggest a route to continual ICL through *iterative compaction*: repeatedly replacing the agent's oldest experiences with compact memory modules as it continues to act (Figure 1). For this process to work, each module must satisfy **compaction quality (R1)**, preserving the knowledge in the experiences it replaces even as new raw experience is appended. Modules must also satisfy **composability (R2)**, so that memories from successive compactions remain useful when served together. Our controlled experiments show that independently trained Cartridges lose effectiveness

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

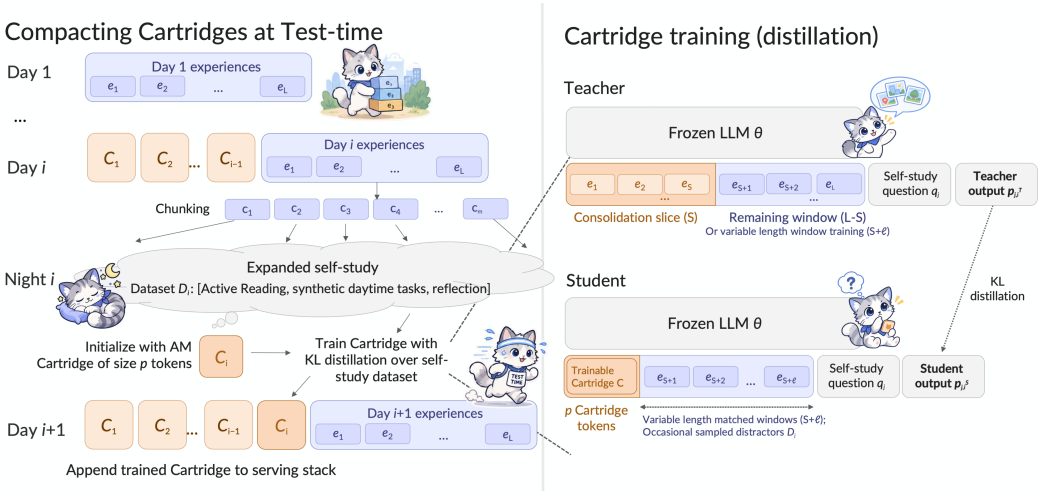


Figure 1: **Continual KV compaction with ACCORDION.** On Night i, experiences from Day i are chunked and used to generate an expanded self-study dataset $\mathcal{D}_i$ to train Cartridge C_i by KL distillation from a full-context teacher model. On Day $i+1$, the frozen C_i joins the KV cache stack alongside new uncompressed context. Our method enables stable multiday compaction and serving, which provides a path towards scaling sleep-time compute for serving with lower cost at test-time and greater agent expertise on the task at hand.

alongside both new context and other memory modules (Section 3). Attention Matching composes more naturally, but retains too little knowledge to achieve high compaction quality.

To meet both requirements, we introduce ACCORDION, which combines the knowledge capacity of trained Cartridges with the composability of Attention Matching (Section 4). The agent alternates between *day*, when it acts and accumulates experience, and *night*, when it consolidates its oldest experiences into a trained KV memory module. Each new module joins the existing memory stack, freeing space for the next day's experience while carrying earlier knowledge forward.

To build each module, the agent revisits the experiences being compacted and generates *self-study examples*, including questions about what it observed and potential future tasks that draw on those experiences. The same model produces answers or proposed actions using focused portions of the original experience, providing targets for self-distillation. We initialize the compact cache with ridge-regularized Attention Matching and compressed RoPE positions, then refine it on these targets. To make the resulting memory useful as experience accumulates, we train it alongside frozen earlier modules and varying amounts of subsequent uncompressed experience. This trains each module in the context in which it will later be used, supporting composition with both earlier memories and new observations. Only the new cache is trained; the base model and earlier modules remain frozen.

We evaluate ACCORDION against ICL, agentic summarization, and training-free KV compaction on three benchmarks covering detailed recall, problem solving under uncertainty, and legal tool use (Section 6). On QuALITY (Pang et al., 2022), after eleven compactions over approximately one million tokens, ACCORDION retains 81% of the accuracy of an ICL baseline given the relevant documents directly. On Blind Spectrum Monitoring from Continual Learning Bench (Asawa et al., 2026), Qwen3.5-9B improves as memories accumulate and reaches a score of 0.358, compared with 0.220 using ICL. It also outperforms the two evaluated frontier models over the final 15 scans, even when they receive the full context. On Harvey LAB (Harvey AI, 2026), ACCORDION passes 87 of 111 legal-task criteria, compared with 73 for an agent with tools but no memory across tasks. It achieves this result with 34% fewer tool calls and 23.3× compaction, the highest compression ratio among the evaluated memory methods. Together, these results show that iterative compaction can preserve useful experience over long trajectories and turn it into better task performance without updating the base model.

Our contributions are as follows:

Identifying what continual KV-cache-based memory requires. We formulate continual in-context learning and identify two requirements for iterative KV compaction: preserving knowledge as new context arrives and combining memories from successive compactions. Controlled experiments reveal where existing methods fall short on each requirement (Section 3).

Building better memories modules with less training. Informed by the failure cases of the existing methods, we combine ridge-regularized Attention Matching, compressed RoPE positions, and self-distillation to build higher-quality KV memory modules. This approach reaches higher accuracy in less training time than standard Cartridges (Section 4.1).

Making memories work together as experience grows. We train each new module alongside frozen earlier memories and varying amounts of subsequent context. This stack-aware training enables ACCORDION to retain and use knowledge across repeated compactions without updating the base model (Section 4.2).

Turning compact memory into better task performance. Across QuALITY, Continual Learning Bench, and Harvey LAB, ACCORDION sustains recall through repeated compactions, improves problem solving over ICL, and strengthens legal tool use. Notably, on Harvey LAB, it achieves these gains with $23.3\times$ compaction and 34% fewer tool calls than an agent without memory across tasks (Section 6).

2 AN ITERATIVE COMPACTION VIEW OF CONTINUAL IN-CONTEXT LEARNING

Let π_θ be a language model agent that encounters a sequence of experiences $e_1, e_2, \ldots$, where each experience may contain a task, the agent's actions, or feedback from the environment. Under typical in-context learning, the agent at step t conditions on its full history $[e_1; \ldots; e_{t-1}]$, which grows without bound as the trajectory continues.

In contrast, iterative context compaction slows this growth by alternating between acting and compacting. The agent acts until its context fills, then compacts its oldest span of consecutive experiences E_i into a memory module M_i much smaller than E_i, and resumes acting. After k such act-compact cycles, the agent conditions on $[M_1; \ldots; M_k; R]$ instead of its full history $[E_1; \ldots; E_k; R]$, where R holds the experiences not yet compacted. We call compaction successful if this substitution never hurts performance on future tasks:

$$\mathbb{E}_{x\sim D}\big[\tau\big(\pi_\theta(\cdot \mid M_1, \ldots, M_k, R, x)\big)\big] \;\geq\; \mathbb{E}_{x\sim D}\big[\tau\big(\pi_\theta(\cdot \mid E_1, \ldots, E_k, R, x)\big)\big] \quad \text{for all } k, \tag{1}$$

where D is the distribution of future tasks and τ scores the agent's output.

Cartridges (Eyuboglu et al., 2026) trains a compact cache end to end with a self-distillation loss, so that the model's output distribution on self-generated questions matches its distribution with the full context in view. Attention Matching (Zweiger et al., 2026) instead fits the cache locally, selecting a subset of the original keys and solving in closed form for per-key biases and values so that each attention head approximates its original attention mass and output on a set of reference queries.

To support repeated cycles of acting and compaction, memory modules must satisfy two requirements: high compaction quality and composability. For compaction quality, each *single* module should faithfully preserve the knowledge contained in the original experiences, including when new raw experiences are appended. For composability, each module should remain effective when used alongside *multiple* other modules.

3 CAN WE ACHIEVE CONTINUAL IN-CONTEXT LEARNING WITH EXISTING KV CACHE COMPACTION METHODS?

To study how existing methods handle learning information sequentially, we set up a simple toy task. CITIES is a recall task where each city maps to a unique combination of industry, terrain, and weather. At each step, we give the agent a paraphrased description of one city, the agent outputs its guess of city name, then the correct name is given as feedback (full details in Appendix A.1.1).

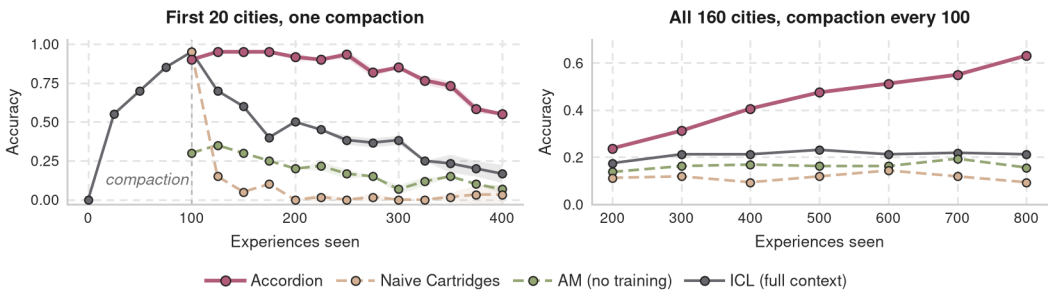


Figure 2: **Existing KV compaction methods fail to retain compacted knowledge on CITIES.** Left measures recall on the first 20 cities after a single $20\times$ compaction. Cartridge recall collapses quickly once new context is appended, Attention Matching has milder degradation but begins with lower retention. Full ICL also forgets as unrelated context accumulates. Right measures accuracy on all 160 cities under repeated compaction every 100 experiences. ACCORDION continues to improve while ICL and baselines plateau.

3.1 CAN WE ACHIEVE HIGH QUALITY SINGLE COMPACTION (R1) WITH EXISTING METHODS?

To study single compaction quality (R1), we have the agent act for 100 experiences on the first 20 cities, compress them by $20\times$ into a memory module, and then act for 300 more experiences about a second, unrelated set of 20 cities.

Cartridges break once long context is appended. After appending only 25 unrelated experiences, we see that Cartridge retrieval accuracy on the first 20 cities drops from 95% to 15% (Figure 2). One failure mode we observe is that Cartridge training optimizes recall over Cartridge contents (first 20 cities) only when the Cartridge is the sole context. This means as soon as uncompressed new context is appended, the model fails to retrieve from the compressed Cartridge.

Attention Matching generalizes better, but at the cost of capacity. After appending 100 experiences, recall accuracy on the AM cache drops by only 10%, while Cartridge recall collapsed to 0% (Figure 2). Attention Matching (AM) yields a compressed KV cache that composes more naturally with later text, since it approximates the attention mass and output of the original text. However, AM fits a layer-wise linear regression against a fixed query set, which restricts its expressivity and causes the training-free AM to under-perform the gradient-based Cartridge. Immediately after compression, Cartridge matches ICL at 95% recall while AM cache reaches only 35% (Figure 2).

3.2 CAN WE ACHIEVE COMPOSABILITY FOR ITERATIVE COMPACTION (R2) WITH EXISTING METHODS?

To study composability (R2), we repeat this act-compress loop: every 100 experiences, which cover 20 new cities, the agent compresses them by $20\times$ into a new memory module and continues acting. We run eight days with seven compactions, for 800 experiences over 160 cities.

Cartridges are not composable knowledge modules. As we stack more Cartridges, recall accuracy stays around 10% (Figure 2), even though it should grow as the agent sees more experiences. Each Cartridge is trained for retrieval on its own, with no other memory modules in context, so stacking them introduces interference it was never optimized for.

Attention Matching is composable but still degrades. Multiple AM caches compose with little interference, much like natural-language text (see also Figure 4). With seven stacked caches, however, accuracy on all 160 cities stays around 15% and does not rise as the agent sees more cities (Figure 2). The plateau comes from ICL itself: AM is built to approximate ICL and stays only slightly below it, and full ICL also plateaus as older cities are forgotten.

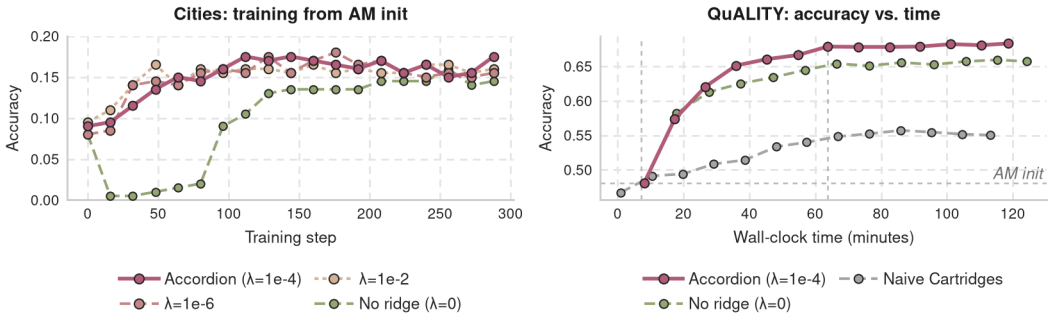


Figure 3: **Ridge-regularized AM initialization stabilizes training and lifts final accuracy.** Left: ridge regularization stabilizes training on the cities task (Llama-3.1-8B). Without it, validation accuracy collapses by step 16 and takes $\sim$90 steps to recover, while every $\lambda \geq 10^{-6}$ reaches the same plateau without collapsing. Right: on QuALITY (Qwen3.5-9B, mean of seven 16-document groups), training from the ridge AM cache reaches 68% in 56 minutes, above untrained AM (48%) and naive Cartridges, which plateau at 56% after 85 minutes.

4 METHOD

4.1 INITIALIZATION FROM RIDGE-REGULARIZED ATTENTION MATCHING

Section 3 showed that Attention Matching (AM) composes with later context but lacks capacity, while trained Cartridges have enough capacity but compose poorly. We combine the two by initializing the Cartridge C_i from an AM cache instead of the first p tokens of the full context, followed by a short training refinement. AM selects the p keys ($\boldsymbol{C}_k$) with the highest attention mass under reference queries $\boldsymbol{q}_1, \cdots, \boldsymbol{q}_n$. It then fits per-key biases $\boldsymbol{\beta}$ to the full cache's attention mass, and solves for values that reproduce its attention output $\boldsymbol{o}(\boldsymbol{q}_i)$ (full formulation in Appendix A.4). We change the value solve in two ways (blue):

$$\boldsymbol{C}_v^\star = \arg\min_{\boldsymbol{C}_v} \sum_{i=1}^{n} \left\| \mathrm{softmax}\left(\tilde{\boldsymbol{q}}_i \tilde{\boldsymbol{C}}_k^\top / \sqrt{d} + \boldsymbol{\beta}\right) \boldsymbol{C}_v - \boldsymbol{o}(\boldsymbol{q}_i) \right\|_2^2 + \lambda \|\boldsymbol{C}_v\|_F^2. \tag{2}$$

(with λ scaled by $\|\tilde{\boldsymbol{X}}\|_2^2$; Appendix A.4) **Rank-mapped positions.** AM keeps each key at its original position, so a stack of AM caches fills the context window as fast as the raw context. We re-rotate the selected keys to positions $1, \ldots, p$, giving $\tilde{\boldsymbol{C}}_k$, and the queries to position $p+1$, giving $\tilde{\boldsymbol{q}}_i$, so each cache occupies only p RoPE positions.

Ridge regularization. We further find that the attention matrix in Eq. 2 is severely ill-conditioned (Figure 10). Without ridge regression, the solve yields value norms $10^{12}\times$ those of real tokens, which is too large for gradient steps to move. The ridge penalty keeps values at natural scale, so that training is numerically stable (Figure 3).

With both changes, training from the AM initialization reaches 68% on QuALITY, above both the untrained AM cache (48%) and naive Cartridges (56%), and passes the latter's plateau in under an eighth of the time (Figure 3).

4.2 CONTINUAL COMPACTION BY EXPANSION AND COMPRESSION

ACCORDION alternates between a *day*, when the agent acts on its stack of memory modules followed by a live window R of raw experience, and a *night*, when the oldest span of R, the *consolidation slice* S_i, is compacted into a new module that replaces it (Figure 1). Each memory module M_i of Section 2 is a *cartridge* C_i, a trainable KV cache of p tokens. A Cartridge should serve the next day's tasks (Equation 1), but the agent cannot observe them while it sleeps. Prior compaction methods build training data from question–answer pairs over a fixed corpus, yet an agent's experience consists of actions, observations, and feedback, and its output is itself an action. Each night therefore first expands S_i into many training examples, then compresses them into C_i.

Expansion by dreaming. During the day, the agent reads each experience once while busy with its task; at night it can revisit S_i as often as compute allows. It samples short *study views* v from S_i and, from each view, proposes new inputs $\tilde{x}$, such as tasks in the daytime format, recall questions, or its own study strategies (Lin et al., 2026). The same frozen model, as the teacher, then answers each input from its view, $\tilde{y} \sim \pi_\theta(\cdot \mid v, \tilde{x})$, forming the dream dataset $\tilde{D}_i$. Because each answer sees only a short, focused view, it avoids the context rot that limits daytime ICL.

Compression by stack-aware distillation. Starting from the AM initialization of Section 4.1, we train C_i where it will be served, behind the frozen stack and ahead of a live window of random length:

$$\mathcal{L}(C_i) = \mathbb{E}_{(v,\tilde{x},\tilde{y})\sim\tilde{D}_i}\, \mathbb{E}_{\ell\sim\mathcal{U}\{0,\ldots,|R|\}} \sum_t D_{\mathrm{KL}}\Big(\pi_\theta(\cdot \mid v, R_\ell, \tilde{x}, \tilde{y}_{<t}) \;\Big\|\; \pi_\theta(\cdot \mid C_{<i}, C_i, R_\ell, \tilde{x}, \tilde{y}_{<t})\Big), \tag{3}$$

where R_ℓ holds the first ℓ experiences of the live window R after S_i, seen by both teacher and student. Conditioning on $C_{<i}$ makes C_i composable with the stack (R2), sampling ℓ matches the window that grows during the day, and, for a fraction of examples, adding distractor experiences that only the student sees keeps C_i from relying on the text that follows it (R1); Appendix B.2 ablates each component.

4.3 ACCORDION ENABLES CONTINUAL IN-CONTEXT LEARNING ON CITIES

Across seven rounds of compaction on CITIES, ACCORDION enables consistent improvement on learning up to 160 cities, reaching 63.1% recall, while the KV compaction and ICL baselines plateau (Figure 2) (R2). On compaction quality (R1), after 300 more experiences are appended, ACCORDION recalls 55% of the compacted cities, against the same baselines below 16% (Figure 2).

5 EXPERIMENTAL SETUP

We evaluate iterative compaction on three settings, each stressing a different real-world difficulty (details in Appendix A.1). **QuALITY** (Pang et al., 2022) tests precise recall over a growing corpus, adding 16 stories each night up to 176 stories total ($\sim$1M tokens). **Continual Learning Bench** (Asawa et al., 2026) tests problem solving under uncertainty, tracking hidden transmitters across 90 noisy scans in **Blind Spectrum Monitoring** and learning an unknown database's structure in **Database SQLite**. **Harvey LAB** (Harvey AI, 2026) tests multi-turn tool use on real legal tasks over a corpus that grows past 12M tokens.

We compare ACCORDION against three main baselines, each representing one way to carry memory continually: (i) ICL, which keeps all past context in the window; (ii) Agentic Context Engineering (ACE) (Zhang et al., 2026), the leading **text-based compaction** method; and (iii) Attention Matching (AM) on its own, the highest-quality **training-free KV compaction** method. Appendix A.5 details our additional baselines KVZip (Kim et al., 2025), Dynamic Cheatsheet (Suzgun et al., 2026), LoRA fine-tuning on the same data (Hu et al., 2022), and naive Cartridges (Eyuboglu et al., 2026).

We test ACCORDION on three attention mechanisms: grouped-query attention (GQA) in Qwen3-32B, sliding-window attention (SWA) in Gemma4-12B-it and Gemma4-31B-it, and Gated DeltaNet (GDN) in Qwen3.5-9B and Qwen3.8-27B. In hybrid architectures, we prepend the Cartridge stack to full-attention layers and train SWA layers the same way. GDN layers instead carry a frozen snapshot of their recurrent state, captured from the context. For each task, we pick a model strong enough to make progress but such that the benchmark does not saturate.

6 RESULTS

6.1 ACCORDION ENABLES PRECISE RECALL OVER LARGE GROWING KNOWLEDGE CORPUS

ACCORDION holds precise details with large capacity. We first test whether ACCORDION can internalize a large, growing corpus, such as a company's proprietary documents. On QuALITY, the agent compacts 16 new stories at $20\times$ each night for eleven nights, reaching 176 stories ($\sim$1M tokens). ACCORDION retains 81% of the single-group ICL ceiling with Qwen3.5-9B and 78% with

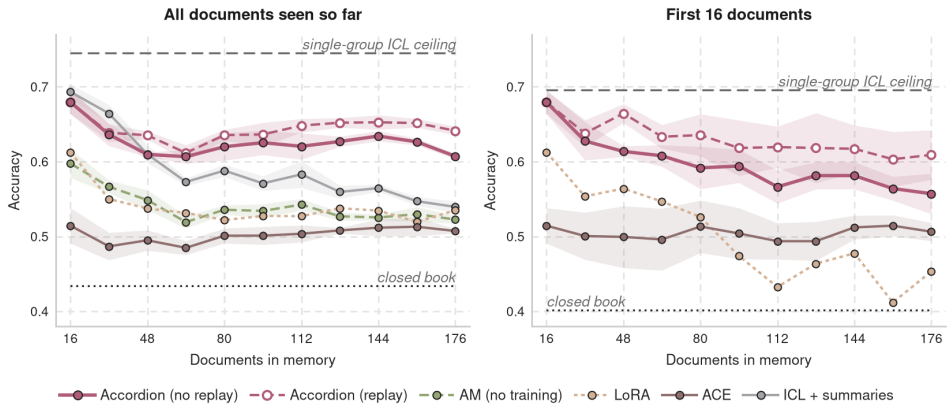


Figure 4: **ACCORDION retains precise recall over 176 stories on QuALITY.** Left: accuracy on all documents seen so far as 20× compression (Qwen3.5-9B). ACCORDION stays at 81% of the single-group ICL ceiling after eleven nights; replaying 10% of earlier stories only adds 3 points (86%). Right: accuracy on the first 16 documents as later ones are added. Frozen Cartridges keep the earliest knowledge, and LoRA forgets steadily.

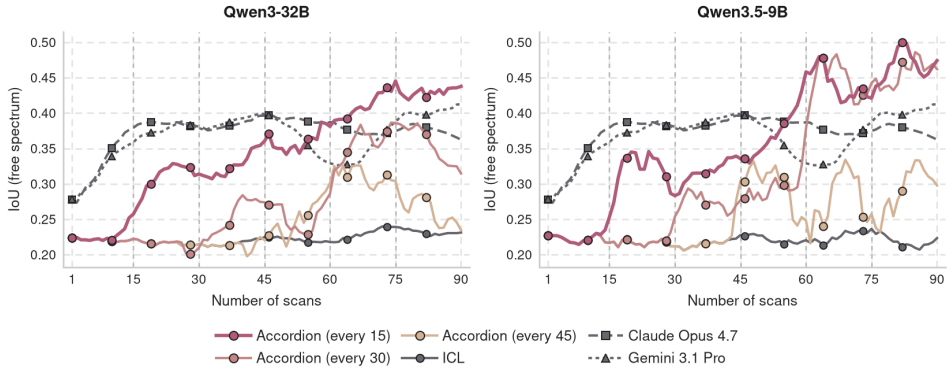


Figure 5: **ACCORDION on Qwen3.5 and Qwen3 overtake frontier model ICL on Blind Spectrum Monitoring.** Blind Spectrum Monitoring measures per-scan IoU over 90 noisy scans (Qwen3-32B left, Qwen3.5-9B right) as benchmark score. Compacting every 15 scans, ACCORDION achieves solid gain over ICL throughout, and reaches 0.430 and 0.468 over the final 15 scans, surpassing Opus (0.376) and Gemini (0.402). Compacting every 30 or 45 scans loses part of the gain.

Gemma4-12B-it (Figure 4), while the strongest baselines, ICL with summaries on Qwen and ACE on Gemma, reach 72% and 71%.

ACCORDION internalizes knowledge well without replay. One common technique to prevent catastrophic forgetting when internalizing large corpora is replay of old knowledge; however, this carries a growing cost over time. ACCORDION freezes earlier Cartridges after training them (Section 4), and does not require replay. Replaying earlier stories, 10% of each stage's training data for Qwen3.5-9B and 20% for Gemma4-12B-it, adds only 3 and 2 points, respectively.

6.2 ACCORDION PRESERVES AND IMPROVES PROBLEM SOLVING UNDER UNCERTAINTY

ACCORDION enables problem solving gains beyond simple recall. Blind Spectrum Monitoring inherently requires inference and problem solving. Remembering every scan fully helps, but does not solve the task. This is because no single scan is reliable due to noisy measurements, and useful signal only emerges over time. ACCORDION improves on ICL by up to 63% with Qwen3.5-9B and Qwen3-32B, while compacting context by 20.3×. Over the final 15 scans, the cartridge stack exceeds both frontier models' ICL performance (Claude Opus 4.7 and Gemini 3.1 Pro) (Figure 5).

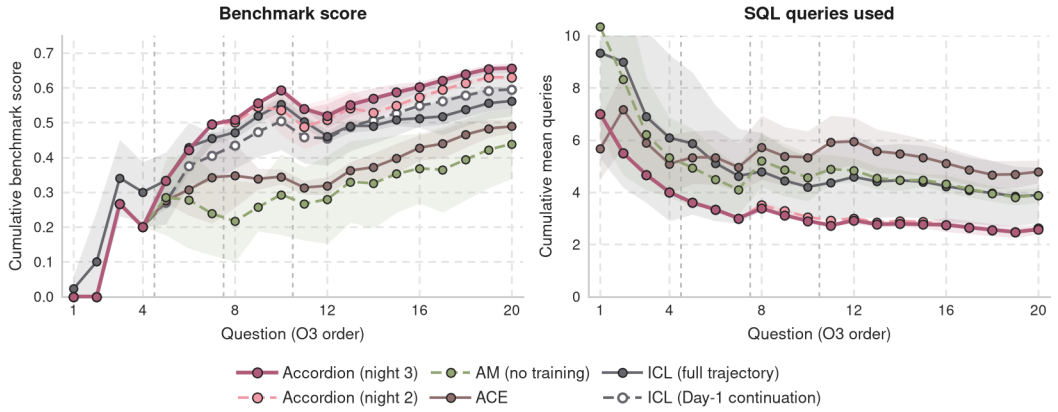


Figure 6: **ACCORDION matches strong ICL baseline on Database SQLite.** Over 20 questions on Database SQLite, ACCORDION on Qwen3.8-27B matches ICL on accuracy (77.5% ACCORDION vs. 71.7% ICL) while using fewer queries on mean (Right), and thus achieves highest benchmark score (Left). Reconstruction-only Attention Matching falls short on accuracy. ACE achieves only slightly higher accuracy and uses the most queries on mean. While ICL is the strongest baseline on short, clean contexts with high uncertainty involved (up to 30K tokens total), ACCORDION helps preserve the problem solving ability and high downstream accuracy of ICL.

Meanwhile on Database SQLite, Qwen3.8-27B ICL is already a strong baseline for problem solving at 71.7% accuracy. Compacting over three nights, ACCORDION matches ICL-level accuracy at 77.5% while cutting mean queries from 3.88 to 2.58, which raises the benchmark score from 56.2 to 65.7. Attention Matching and ACE (Zhang et al., 2026) score 54.3 and 52.4 (Table 9).

ACCORDION compacts learning signal rather than raw experience. We trace the gains of ACCORDION to several factors through ablations. First, **each added Cartridge extracts more signal**. Because the same model serves as teacher and student, any gain on BSM must come from signal extracted during study. On BSM, the gap to ICL opens with the first Cartridge and widens as more are stacked, each built from a further round of study (Figure 5). Compacting every 15 scans, which stacks five Cartridges over the run, outperforms every 30 or 45, which stack two or one. We attribute the gain to both the compaction cadence and the additional study each Cartridge adds (Appendix B.3.1). **Sufficient capacity** matters as well. On Database, we similarly see that a larger Cartridge size than recall tasks boosts performance (Appendix B.4). Lastly, **latent KV retains studied signal over text.** On BSM with Qwen3-32B, serving a text summary of the same self-study data scores 0.28 IoU, compared to 0.345 for ACCORDION (Appendix B.3.2).

6.3 ACCORDION YIELDS STRONG AGENTIC PERFORMANCE ON COMPLEX LEGAL TASKS

Harvey LAB (Harvey AI, 2026) is our most realistic setting. Eighteen legal matters arrive one at a time in a law firm's document store, and the agent solves each step's tasks with shell tools under a 40-turn cap, graded on 111 rubric criteria. The agent keeps tool access to the full store, which grows past 12M tokens, and within an answer every method sees its own tool calls, so methods differ only in what they carry between answers. Carrying the full history is impossible: the agent's own transcripts alone reach up to 294K tokens, past the model's 262K-token window. The tools-only baseline therefore carries nothing between answers and must rediscover the store at every step. ACCORDION instead compacts each step's transcript, the agent's tool calls and their results, into a Cartridge at 23.3× (Appendix A.2.6); the document store itself is never compacted. Across 18 matters, ACCORDION passes 87 of 111 criteria with Qwen3.8-27B and 69 with Gemma4-31B-it, against 73 and 54 for the tools-only baseline, reaching 88% and 75% of the oracle ceiling (Table 1, Figure 7).

ACCORDION internalizes the setting. The gain comes with a change in behavior. With ACCORDION, the Qwen3.8-27B agent makes about a third fewer tool calls than the control, and fewer than

Table 1: **Harvey LAB, 18 matters, Qwen3.8-27B.** Compaction ratio, accuracy (fraction of 111 rubric criteria passed), and average tool calls per answer.

Method	Compaction	Accuracy	Tool calls
Tools only (no cross-step memory)	–	0.658	16.4
KVzip (Kim et al., 2025)	4×	0.586	12.9
ACE (Zhang et al., 2026)	11.0×	0.667	14.4
ACCORDION (ours)	**23.3**×	**0.784**	**10.8**
Oracle retrieval	–	*0.892*	–

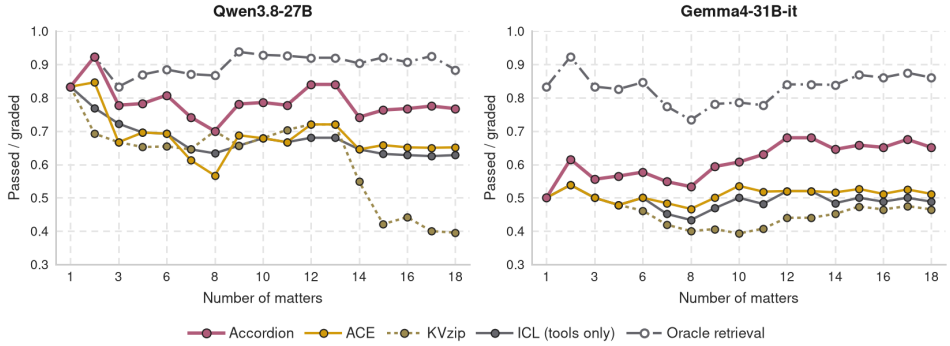


Figure 7: **ACCORDION carries the agent's experience across answers, reaching up to 88% of the oracle ceiling as the store grows past 12M tokens.** Criteria passed over criteria graded in a trailing window of five arrival steps (Qwen3.8-27B left, Gemma4-31B-it right). Tools only has tool access to the full store and no memory across answers; oracle retrieval (dashed) also places the matter's documents in context. Compacting each step's transcripts at 23.3×, ACCORDION passes 87 and 69 of 111 criteria, 88% and 75% of the oracle, against 73 and 54 for tools only.

with any other memory (Table 1). Agents without memory tend to reach for tools even when the answer is already known (Qian et al., 2025), and experience memory has been shown to shorten trajectories (Wang et al., 2025b; Ouyang et al., 2026). We read the fewer calls as internalization: the agent no longer rediscovers directory layouts, matter numbers, or dead-end searches, and goes more directly to the documents it needs, while holding the most compact memory of any method. Keeping more of the past does not substitute for this. KVzip retains a larger share of the same transcripts without training and still trails ACCORDION by a wide margin on both models, which suggests that what the memory holds matters more than how much of the experience it keeps.

Sleep-time study pays off at test time. ACCORDION moves work from answering into sleep (Lin et al., 2025b). Each night the agent reviews its own trajectories, questions them field by field, and distills the answers into memory (Section 4.2); the next day it answers more accurately and in fewer steps. Because each Cartridge is reused at every later step, this cost is paid once per night rather than once per query. How the study is stored matters too: ACE learns from the same trajectories but keeps its lessons as a text playbook, and it barely improves on the control, while ACCORDION gains on both models.

7 CONCLUSION

As we move towards a world in which large language models are increasingly agentic, operate over long horizons, and interact with users and real-world environments over extended periods of time, context management and continual improvement from experience become increasingly important. The naive approach of putting experiences in context is limited, and existing compression approaches like Attention Matching and independently trained cartridges only offer partial solutions. As such, ACCORDION offers a first step towards full continual in-context learning.

AI USE STATEMENT

We used generative AI tools to generate synthetic datasets and for research ideation and execution, but not to formulate or prove mathematical claims; the other tasks requiring disclosure do not apply. We also used them to search the literature, draft parts of the paper, and edit it for readability. Synthetic self-study data are part of ACCORDION itself (Section 4.2), and the Database SQLite question order was generated with Claude Fable. We have reviewed all AI-assisted work, including every citation, and take responsibility for the final content of this work.

REFERENCES

Ekaterina Alimaskina, Denis Shveykin, Gleb Molodtsov, Igor Shalygin, Alexey Kadeishvili, and Aleksandr Beznosikov. Self-study reconsidered: The hidden fragility of learning from self-generated qa, 2026. URL https://arxiv.org/abs/2606.32002.

Parth Asawa, Christopher M. Glaze, Gabriel Orlanski, Ramya Ramakrishnan, Benji Xu, Asim Biswal, Vincent Sunn Chen, Frederic Sala, Matei Zaharia, and Joseph E. Gonzalez. Continual learning bench: Evaluating frontier AI systems in real-world stateful environments. *arXiv preprint arXiv:2606.05661*, 2026. doi: 10.48550/arXiv.2606.05661. URL https://arxiv.org/abs/2606.05661.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. Nested Learning: The Illusion of Deep Learning Architectures. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-1565. URL https://papers.nips.cc/paper_files/paper/2025/hash/4309616aaed8e848009bc4a7ef73b493-Abstract-Conference.html.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-3786.

Ali Behrouz, Farnoosh Hashemi, Adel Javanmard, and Vahab Mirrokni. Language models need sleep: Learning to self-modify and consolidate memories, 2026. URL https://arxiv.org/abs/2606.03979.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Lucas Caccia, Alan Ansell, Edoardo Ponti, Ivan Vulić, and Alessandro Sordoni. Training plug-n-play knowledge modules with deep context distillation. In *Conference on Language Modeling (COLM)*, 2025. URL https://arxiv.org/abs/2503.08727.

Vivek Chari, Guanghui Qin, and Benjamin Van Durme. KV-Distill: Nearly lossless learnable context compression for LLMs, 2025. URL https://arxiv.org/abs/2503.10337.

Tong Chen, Hao Fang, Patrick Xia, Xiaodong Liu, Benjamin Van Durme, Luke Zettlemoyer, Jianfeng Gao, and Hao Cheng. Generative adapter: Contextualizing language models in parameters with a single forward pass. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bc3sUsS6ck.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3829–3846. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.232. URL https://aclanthology.org/2023.emnlp-main.232/.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In *ECAI 2025 – 28th European Conference on Artificial Intelligence*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pp. 2993–3000. IOS Press, 2025. doi: 10.3233/FAIA251160.

Chuheng Du, Junyi Chen, Hanlin Tang, Kan Liu, Tao Lan, Lin Qu, Chaoyue Niu, Shengzhong Liu, Guihai Chen, and Fan Wu. C^2KV: Compressed and composable KV cache reuse for efficient LLM inference. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026. doi: 10.1145/3770855.3817715. URL https://arxiv.org/abs/2607.17715.

Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Atri Rudra, James Y Zou, Azalia Mirhoseini, and Christopher Re. Cartridges: Lightweight and general-purpose long context representations via self-study. In *International Conference on Learning Representations*, volume 2026, pp. 42642–42687, 2026.

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 *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/0b276510ec2d3f6613a8b60c41ff0438-Abstract-Conference.html.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. Model editing at scale leads to gradual and catastrophic forgetting. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515, 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Momchil Hardalov, Gonzalo Iglesias, and Adrià de Gispert. Cartridges at scale: Training modular KV caches over large document collections. *arXiv preprint arXiv:2606.04557*, 2026. doi: 10.48550/arXiv.2606.04557. URL https://arxiv.org/abs/2606.04557.

Harvey AI. Harvey lab: The legal agent benchmark, 2026. URL https://github.com/harveyai/harvey-labs/tree/v1.1.0. Announcement: https://www.harvey.ai/blog/introducing-harveys-legal-agent-benchmark.

Kelly Hong, Anton Troynikov, and Jeff Huber. Context rot: How increasing input tokens impacts LLM performance. Technical report, Chroma, 2025. URL https://research.trychroma.com/context-rot.

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.

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with LoRA, 2023.

Jang-Hyun Kim, Junyoung Yeom, Sangdoo Yun, and Hyun Oh Song. Compressed context memory for online language model interaction. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/97dc07f1253ab33ee514f395a82fa7cc-Abstract-Conference.html.

Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W. Lee, Sangdoo Yun, and Hyun Oh Song. KVzip: Query-agnostic KV cache compression with context reconstruction. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5585.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

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*, pp. 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.353.

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*, volume 37, 2024. doi: 10.52202/079017-0722.

Zeju Li, Yizhou Zhou, and Qiang Xu. Latent context compilation: Distilling long context into compact portable memory, 2026. URL https://arxiv.org/abs/2602.21221.

Jessy Lin, Luke Zettlemoyer, Gargi Ghosh, Wen-Tau Yih, Aram Markosyan, Vincent-Pierre Berges, and Barlas Oğuz. Continual Learning via Sparse Memory Finetuning, 2025a. URL https://arxiv.org/abs/2510.15103.

Jessy Lin, Vincent-Pierre Berges, Xilun Chen, Wen-tau Yih, Gargi Ghosh, and Barlas Oğuz. Learning facts at scale with active reading. In *International Conference on Learning Representations*, 2026.

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, 2025b. URL https://arxiv.org/abs/2504.13171.

Yujian Liu, Jiabao Ji, Li An, Rohit Jain, Gungor Polatkan, Siyu Zhu, and Shiyu Chang. Practical online KV cache compaction for LLM agents: An empirical study, 2026. URL https://arxiv.org/abs/2608.00902.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f87522788a2be2d171666752f97ddebb-Abstract.html.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3d77c6dcc7f143aa2154e7f4d5e22d68-Abstract-Conference.html.

Charles O'Neill, Alex Sandomirsky, Harry Partridge, Mudith Jayasekara, and Max Kirkby. Still: Amortized KV cache compaction in a single forward pass. *arXiv preprint arXiv:2606.07878*, 2026. doi: 10.48550/arXiv.2606.07878. URL https://arxiv.org/abs/2606.07878.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.25140.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. doi: 10.48550/arXiv.2310.08560. URL https://arxiv.org/abs/2310.08560.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel R. Bowman. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5336–5358, 2022.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Cheng Qian, Emre Can Acikgoz, Hongru Wang, Xiusi Chen, Avirup Sil, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. SMART: Self-aware agent for tool overuse mitigation. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. URL https://aclanthology.org/2025.findings-acl.239/.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by Distilling Context, 2022. URL https://arxiv.org/abs/2209.15189.

Weihang Su, Yichen Tang, Qingyao Ai, Junxi Yan, Changyue Wang, Hongning Wang, Ziyi Ye, Yujia Zhou, and Yiqun Liu. Parametric retrieval augmented generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025. doi: 10.1145/3726302.3729957.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7080–7106. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-long.333. URL https://aclanthology.org/2026.eacl-long.333/.

Arnuv Tandon, Karan Dalal, Xinhao Li, Daniel Koceja, Marcel Rød, Sam Buchanan, Xiaolong Wang, Jure Leskovec, Sanmi Koyejo, Tatsunori Hashimoto, Carlos Guestrin, Jed McCaleb, Yejin Choi, and Yu Sun. End-to-End Test-Time Training for Long Context, 2025. URL https://arxiv.org/abs/2512.23675.

Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*, 639: 130193, 2025a. doi: 10.1016/j.neucom.2025.130193.

Xun Wang, Ruishuo Chen, Zhuoran Li, Yu Chen, and Longbo Huang. When context returns: Toward robust internalization in on-policy distillation, 2026. URL https://arxiv.org/abs/2606.11627.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, Jingbo Shang, and Julian McAuley. MEMORYLLM: Towards self-updatable large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 50453–50466. PMLR, 2024. URL https://proceedings.mlr.press/v235/wang24s.html.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025b.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5e5fd18f863cbe6d8ae392a93fd271c9-Abstract-Conference.html.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candès, and Tatsunori Hashimoto. Synthetic continued pretraining. In *International Conference on Learning Representations*, 2025.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. CacheBlend: Fast large language model serving for RAG with cached knowledge fusion. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys)*, pp. 94–109, 2025. doi: 10.1145/3689031.3696098.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. In *International Conference on Learning Representations*, 2025.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, 2026.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1506.

Ziyang Zheng, Zeju Li, Xiangyu Wen, Jianyuan Zhong, Junhua Huang, Lei Chen, Mingxuan Yuan, and Qiang Xu. Context distillation as latent memory management, 2026. URL https://arxiv.org/abs/2605.28889.

Adam Zweiger, Jyothish Pari, Han Guo, Yoon Kim, and Pulkit Agrawal. Self-adapting language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.neurips.cc/paper_files/paper/2025/hash/6b41e04c41726e2a60e456d0a2b961ab-Abstract-Conference.html.

Adam Zweiger, Xinghong Fu, Han Guo, and Yoon Kim. Fast KV compaction via attention matching. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

APPENDIX

A Implementation details **15**

A.1 Tasks 15

A.2 Nightly data pipeline 17

A.3 Hyperparameters 19

A.4 Attention Matching initialization 21

A.5 Baselines 22

B Additional results **24**

B.1 Attention Matching initialization 24

B.2 QuALITY ablations 25

B.3 Blind Spectrum Monitoring ablations 25

B.4 Database SQLite ablations 26

C Related work **27**

D Limitations **28**

This appendix first provides reproducibility details for experiments: task definitions, the nightly data pipeline, hyperparameters, the Attention Matching initialization, and baselines (Appendix A). It then reports the supporting analyses referenced in the main text (Appendix B), related work (Appendix C), and limitations (Appendix D).

A IMPLEMENTATION DETAILS

A.1 TASKS

A.1.1 CITIES

CITIES is a synthetic recall task generated deterministically, without an LLM. Each city is a unique (industry, terrain, weather) triple with a two-word name, and every city shares two of its three attributes with at least one other city, so the full triple is needed to identify it. At each step the agent reads a paraphrased description of one city, answers with a city name in `\boxed{}`, and receives the correct name as feedback. One (question, answer, feedback) triple is an *experience*.

All methods consume the same fixed stream of experiences, acted once at temperature 0.7. The ICL baseline sees the same observations but produces its own answers. We score exact match of the first `\boxed{}` answer under greedy decoding, on one held-out paraphrase per city. All CITIES results use Llama-3.1-8B-Instruct. Table 2 summarizes the task and Figure 8 shows one experience.

```
[user]      Observation: What is the name of the city with the following description:
            The landscape is swamp, featuring boggy ground and slow-moving water.
            It is foggy with thick mist, and the conditions are explicitly classified
            as foggy. Resource operations and freight routes define much of the
            local industry.
[assistant] Action: \boxed{Misty Evergreen}
[user]      Feedback: Your answer is incorrect. The correct answer was
            \boxed{Starlit Summit}.
```

Figure 8: One CITIES experience from day 1.

Table 2: **CITIES at a glance.** Token counts use the Llama-3.1 tokenizer.

Cities	200
Possible attribute triples ($7 \times 9 \times 5$)	315
Paraphrases per city (train / validation)	25 / 1
Description length	34–56 tokens
Experience length (chat turn)	≈100 tokens
Experiences per day (new cities per day)	100 (20)
Stream length	1,000 experiences (10 days; R2 uses the first 8)
Live window L / consolidation slice S	200 / 100 experiences
Trainable Cartridge slots per night	512
Compression	$\approx 19\times$
System prompt (frozen in the first Cartridge)	331 tokens

Single compaction (R1, Figure 2). Experiences 1–100 (cities 0–19) are compacted once. At $x = 100$ the Cartridge is alone in context; for $100 < x \leq 200$, experiences 101 to x follow it; beyond $x = 200$ the agent keeps acting on further paraphrases of cities 20–39. We score the 20 questions of cities 0–19.

Iterative compaction (R2, Figure 2). Every 100 experiences, the oldest 100 experiences of the 200-experience window are compacted into a new Cartridge, for 7 nights. At step x the model sees the stack $C_1, \ldots, C_k$ followed by the live experiences; ICL sees all x experiences as raw text. We score 160 questions (cities 0–159), so the attainable maximum at step x is $x/800$.

A.1.2 QUALITY

We use QuALITY v1.0.1 (Pang et al., 2022) and number its articles by first appearance in the dev file, then the train file. The stream consists of 11 groups of 16 consecutive articles: groups g_0–g_6 are dev articles 0–111 and g_7–g_{10} are train articles 115–178. One group is compacted per night, for 176 articles and 1.05M tokens after 11 nights (Table 3). Every question of a group's articles is evaluated; training uses only self-study data. QuALITY has no acting phase.

Table 3: **QuALITY groups.** Tokens use the Qwen3.5 tokenizer. p is the number of Cartridge slots at $20\times$ compression, $p = \lfloor \text{tokens}/20/16 \rfloor \cdot 16$.

	g_0	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	total
questions	289	280	289	286	297	292	299	263	276	262	265	3,098
tokens (k)	104.1	100.3	105.2	104.6	63.0	107.2	75.0	108.2	99.1	95.8	89.1	1,051.6
p, Qwen3.5	5200	5008	5248	5216	3136	5360	3744	5408	4944	4784	4448	52,496
p, Gemma4	5200	5008	5264	5248	3152	5360	3760	5424	4944	4784	4448	52,592

Scoring. No article is in context at test time; the Cartridge stack is the only memory. We extract the letter with the answer parser of Zweiger et al. (2026); an unparseable answer counts as wrong. Cartridge methods are decoded greedily (at most 64 new tokens) inside the trainer.

Metric and bounds. After night k we report question-weighted accuracy over all questions of $g_0, \ldots, g_k$, from 289 questions after g_0 to 3,098 after g_{10} (standard error ≈ 0.029 and ≈ 0.010). We use three bounds. The *single-group ICL ceiling* places each group's 16 articles alone in context and scores that group's questions (0.745 for Qwen3.5-9B, 0.725 for Gemma4-12B-it). *Closed book* uses no context (0.434 / 0.456). *Joint training* trains one Cartridge with all slots for 2 epochs on the pooled self-study rows of all groups, starting from the untrained AM stack (0.660 / 0.618).

A.1.3 BLIND SPECTRUM MONITORING

Blind Spectrum Monitoring comes from Continual Learning Bench (Asawa et al., 2026). The agent receives 90 noisy scans of a shared radio band, one at a time. After each scan it reports every transmitter it believes exists, with its center frequency, bandwidth, power, and whether it is active

or dormant. Transmitters fade in and out, and some stay dormant for many scans, so no single scan reveals the full population.

Each scan is scored by the intersection over union (IoU) between the spectrum the agent declares free and the spectrum that is truly free; we report the mean IoU over all 90 scans. Evaluation is causal: a scan sees the Cartridges built from earlier scans, followed by the scans that arrived since the last compaction. Compacting every 15, 30, or 45 scans ends the run with five, two, or one Cartridges. We run Qwen3.5-9B and Qwen3-32B, and compare to Claude Opus 4.7 and Gemini 3.1 Pro with the full trajectory in context.

A.1.4 DATABASE SQLITE

We build on the Database SQLite task of Continual Learning Bench and reformat its prompts: each turn's prompt is shorter, and past context is clearly labeled. This gives medium-sized ($\sim$30B) open models a usable ICL baseline, both stateless and stateful. We replace the original benchmark ordering with one generated by Claude Fable so that later questions can build on what earlier ones reveal, which maximizes the continual learning signal over the 20 questions; every method, including all baselines, uses this same ordering. We use a four-day compaction schedule, with compaction after questions 4, 7, and 10. For Qwen3.8-27B, we sample at temperature 0.7 with native thinking disabled and 1–5 sentences of explicit reasoning. We use the first 20 questions of the benchmark, which share one ground-truth database; tasks whose database changes over time are outside the scope of this work.

A.1.5 HARVEY LAB

We use the firm-knowledge tasks of Harvey LAB (Harvey AI, 2026). Eighteen legal matters arrive one at a time into a law firm's document store, which grows past 12M tokens. The store is reached only through tools; no arm holds it in context. What the memory arms compact is the agent's own transcripts, which themselves exceed the model's window by the final steps. At arrival step k the store holds matters $1, \ldots, k$, and the agent answers the tasks scheduled at that step. A task may concern any matter that has arrived, not only the newest one. The schedule has 42 answers over 17 steps.

The agent acts with six tools, `bash`, `read`, `write`, `edit`, `glob`, and `grep`, all available without restriction, under a 40-turn cap and a 60K-token budget per request, and writes its deliverable to a file. Every arm shares this harness and differs only in the memory attached when it answers. We run Qwen3.8-27B and Gemma4-31B-it at temperature 0.7.

Each answer is graded on a subset of its task's rubric criteria, 111 criteria in total. Claude Sonnet gives one pass or fail verdict per criterion from the saved deliverable, and we report the fraction passed. Tool calls in Table 1 are the mean number of agent turns per answer.

A.2 NIGHTLY DATA PIPELINE

A.2.1 GENERAL SETUP

In the growing-stack experiments, during the day the agent acts and appends experiences to a first-in-first-out live window of L experiences. Each night it compacts the oldest S experiences, the consolidation slice S_i, into a new Cartridge of $p = \lfloor |S_i|/r/16 \rfloor \cdot 16$ slots for compression ratio r, and the next day starts from the remaining window. The base model stays frozen; only the newest Cartridge is trained, and all earlier Cartridges are frozen.

Task dreaming. The model writes study tasks conditioned on a *study view*: one chunk of S_i, all of S_i with one chunk marked as the focus, or all of S_i. Tasks either re-ask observed daytime tasks or are newly written from the view, with a genre sampled per task. A *consolidation fraction* sets the share of tasks that target S_i rather than the remaining window.

Action dreaming and targets. The same frozen model, with the study view in context, answers each task, and we store its top-20 log-probabilities for every answer token. The student minimizes $-\sum_{j \le 20} p_T(j) \log p_S(j)$ over the teacher's top-20 tokens at temperature 1. We never train on hard labels.

Student context. Each row uses one of three contexts behind the Cartridge stack: *matched window*, where teacher and student both see the first $\ell \sim \mathcal{U}\{0, \ldots, |R|\}$ remaining experiences; *Cartridge only* ($\ell = 0$); and *distractor*, where the student additionally sees $k \sim \mathcal{U}\{0, \ldots, L - S\}$ experiences that the teacher does not. The mixture weights are task-specific. On QuALITY, a *distractor document* is placed at the head of every training pack and a fixed fraction of rows attend to it; it receives no loss. *Replay* replaces a fraction of a night's rows with rows from earlier nights, keeping the total fixed.

Optimization. We train with Adam. Unless specified otherwise, the learning rate warms up linearly and then decays by cosine to a tenth of its peak. The AM attention biases stay frozen. On Qwen3.5 the recurrent GatedDeltaNet state is frozen as well, while on Gemma4 the sliding-window state is trained with its own learning rate.

A.2.2 CITIES

Each night the model writes about 2,000 new paraphrase questions: 75% about cities in the slice and 25% about the remaining window. The teacher answers each question greedily after reading the slice and, for paired rows, part of the remaining window. We keep a question only if the teacher's answer agrees with its own earlier attempt at that experience, which removed 146 of 2,300 rows on the R1 night. The student never sees the system prompt, which is stored in C_1. There is no replay. Table 4 lists the resulting mixture.

Table 4: **Nightly CITIES data.** Row fractions after filtering. Distractors are the chronological prefix of the remaining window.

Row type (target)	Teacher sees	Student sees	R2	R1
Distractor (slice)	S_i	k remaining	0.568	0.569
Matched window (slice)	S_i + ℓ remaining	ℓ remaining	0.199	0.107
Cartridge only (slice)	S_i	nothing	0	0.074
Remaining window	S_i + ℓ remaining	ℓ remaining	0.233	0.249
Rows per night			2,101–2,221	2,154

The naive Cartridge trains only on slice questions in Cartridge-only mode, with the system prompt kept as live text. R2 dream data was generated once from the fixed stream and reused across arms.

A.2.3 QUALITY

Each night consolidates one group, so no live window follows the new Cartridge; a distractor document plays the role of later context. Each model writes its own study data, 128 rows per article (2,048 per group).

A study view is the whole article, with a 4,000-character focus passage in 75% of rows. Questions come in four equally weighted genres: structuring, summarization, question, and use case. The same model answers with the view in context (temperature 0.7 for Qwen with thinking disabled, 1.0 for Gemma). Half of the rows place the first article of the next group before the question as a distractor. Replay arms replace 10% (Qwen) or 20% (Gemma) of the new rows with rows sampled uniformly from earlier groups.

A.2.4 BLIND SPECTRUM MONITORING

Each night the agent studies the scans since the last compaction in digestible pieces, one question family per skill: auditing a report for transmitters counted twice, finding persistent transmitters missing from the current scan, separating one-off detections from recurring ones, refining a transmitter's center and width from repeated sightings, recalling what a given stretch of scans showed, and writing a report in the required format. The same model answers each question from a short, focused view of the scans, and distillation aggregates the answers into the Cartridge (Section 4.2).

A.2.5 DATABASE SQLITE

We use two self-study strategies. *Synthetic tasks* turn lessons from the day's query results and feedback into new questions in the daytime format. The same model reads the saved experience and writes the first SQL query it would run, without executing it. *Active Reading* asks the model to propose study strategies for a focused experience, then apply them to write short study documents. We distill the generated answers and study documents using the teacher's top-20 probabilities.

A.2.6 HARVEY LAB

The Cartridge for step k compacts the agent's own earlier work, not the matter documents. Its source is the tools-only transcripts of the step that just finished; documents and tasks enter only as fragments the agent itself read. Earlier matters are not re-read: the chain is conditioned, so the Cartridge for step k is built on top of the one for step $k-1$ and inherits what that already holds. KVzip compresses the same step, so both arms compact the same material, one by distillation and one by eviction (at $23.3\times$ and $4\times$, respectively). Compaction ratios on Harvey are therefore relative to the rendered transcript of the step, not to the document store. Each earlier answer is rendered as one chunk: the step and task, then the agent's turns in order, with each tool call, its arguments truncated to 400 characters, and its result truncated to 4,000 characters.

Questions. Each step dreams 150 questions, split evenly across chunks with at least three per chunk. Questions are sampled at temperature 0.8; we keep those of 15 to 300 characters and drop duplicates.

Teacher and student rows. The teacher is the same model without tools. It sees that step's rendered transcripts, capped at the most recent 120K characters, then the source chunk, then the question, and we record the top-20 log-probabilities of its answer as the distillation target. The student sees only the first 4,000 characters of one chunk and the question. In one row of four that chunk is the true source; in the other three it is a different answer from the same step, chosen by a deterministic interleave. The Cartridge must therefore carry what the fuller history gave the teacher. Separately, half of the training packs carry an unrelated matter's document at their head, which receives no loss, so the Cartridge is also trained to answer with foreign context in front of it.

A.3 HYPERPARAMETERS

A.3.1 CITIES AND QUALITY

Table 5 lists the configuration of ACCORDION on CITIES and QuALITY. All runs use ridge AM initialization with rank-mapped positions and $\lambda = 10^{-4}$ (Appendix A.4), train keys and values, and keep the attention biases frozen.

Table 5: **ACCORDION hyperparameters.** R1/R2: the CITIES settings of Appendix A.1.1. SWA: Gemma4 sliding-window state. $^{\dagger}5 \times 10^{-4}$ for night g_6 of the Gemma no-replay chain.

	CITIES R2	CITIES R1	QuALITY	QuALITY
Base model	Llama-3.1-8B-Instruct	Llama-3.1-8B-Instruct	Qwen3.5-9B	Gemma4-12B-it
Cartridge layers (KV heads, head dim)	32 (8, 128)	32 (8, 128)	8 full-attn. (4, 256)	8 full-attn. (1, 512)
Slots per night	512	512	3,136–5,408	3,152–5,424
Compression	$\approx 19\times$	$\approx 19\times$	$20\times$	$20\times$
AM reference queries per KV head	33,616	33,616	20,480	20,480
Peak LR, values / keys	0.02 / 0.02	0.02 / 0.02	0.02 / 0.02	0.02 / 10^{-4}
Peak LR, SWA state	–	–	–	$10^{-3\dagger}$
Warmup steps	16	16	16	32
Packs per step $\times$ pack tokens	$4 \times 12{,}288$	$2 \times 12{,}288$	$1 \times 8{,}192$	$1 \times 8{,}192$
Epochs per night	2	2	2	2

A.3.2 BLIND SPECTRUM MONITORING

Table 6 lists the BSM configuration in the format of Table 5; both models use the same settings. Each night adds a 768-slot Cartridge initialized with ridge AM and rank-mapped positions over the last 45K characters of the night's scans (Appendix A.4).

Table 6: **ACCORDION hyperparameters on BSM.**

	BSM	BSM
Base model	Qwen3.5-9B	Qwen3-32B
Attention	GDN hybrid	dense
Slots per night	768	768
Compression	$20.3\times$	$20.3\times$
AM ridge λ	10^{-4}	10^{-4}
AM probe tokens	8,192	8,192
AM init window	45K chars	45K chars
Self-study questions per night	96	96
Peak LR, values / keys	10^{-2} / 10^{-2}	10^{-2} / 10^{-2}
Warmup steps	8	8
Pack tokens	8,192	8,192
Epochs per night	12	12
Optimizer steps per night	≈ 423	≈ 423

A.3.3 DATABASE SQLITE

Table 7 lists the database configuration in the format of Table 5. The memory uses AM initialization and trains keys and values while keeping attention biases frozen. This historical run does not preserve GDN recurrent state across compaction.

Table 7: **ACCORDION hyperparameters on Database SQLite.**

	Database SQLite
Base model	Qwen3.8-27B
Cartridge layers (KV heads, head dim)	16 full-attn. (4, 256)
Total slots per night	1,885
Trainable slots	1,538
Overall compression	$5.41\times$
AM ridge λ	10^{-2}
Peak LR, values / keys	0.003 / 0.003
Warmup steps	0 (constant LR)
Packs per step $\times$ pack tokens	$1 \times 1{,}280$
Optimizer steps per night	120

A.3.4 HARVEY LAB

Table 8 lists the Harvey configuration in the format of Table 5. Each step adds a 2,048-slot Cartridge initialized with ridge AM and rank-mapped positions, $\lambda = 10^{-4}$, over the last 120K characters of the rendered past (Appendix A.4). Both models train for four epochs per step; Gemma takes fewer optimizer steps because its agent takes fewer turns per answer (about 11 against 16), so each step yields fewer packs.

Table 8: **ACCORDION hyperparameters on Harvey.** SWA: Gemma4 sliding-window state.

	Harvey	Harvey
Base model	Qwen3.8-27B	Gemma4-31B-it
Slots per step	2,048	1,024
Compression	23.3×	19.1×
Peak LR, values / keys	4×10^{-3} / 2×10^{-3}	0.02 / 10^{-4}
Peak LR, SWA state	–	10^{-3}
Warmup steps	5	32
Pack tokens	16,384	16,384
Epochs per step	4	4
Optimizer steps per step	≈ 105	≈ 88
Distractor packs	50%	50%

A.4 ATTENTION MATCHING INITIALIZATION

Attention Matching. Attention Matching (AM) (Zweiger et al., 2026) approximates each attention head's KV cache $(\boldsymbol{K}, \boldsymbol{V}) \in \mathbb{R}^{T \times d}$ with a compact cache $(\boldsymbol{C}_k, \boldsymbol{\beta}, \boldsymbol{C}_v)$ of $p \ll T$ entries, where $\boldsymbol{\beta} \in \mathbb{R}^p$ is a per-key scalar attention bias. Given reference queries $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$ obtained from a prefill over some chosen contexts, AM keeps the p keys receiving the highest attention mass as $\boldsymbol{C}_k$. It then fits $\boldsymbol{\beta}$ so the compact cache reproduces the full cache's attention *mass*, and $\boldsymbol{C}_v$ so it reproduces the attention *output*, over the reference queries. Both fits are linear solves, so compaction takes minutes. Our reference queries come from two sources: the slice's own tokens, and the questions of 256 self-study conversations about the slice.

Compressing RoPE positions. AM leaves each kept key at its original RoPE position, so the compacted context still occupies T positions of the model's window. We instead re-rotate the j-th kept key from its original position n_j to position j, and place the reference queries, captured before RoPE, at position $p+1$, immediately after the last key:

$$(\tilde{\boldsymbol{C}}_k)_j = R_j R_{n_j}^{-1} (\boldsymbol{C}_k)_j, \qquad \tilde{\boldsymbol{q}}_i = R_{p+1}\, \bar{\boldsymbol{q}}_i, \tag{4}$$

where R_m is the RoPE rotation at position m and $\bar{\boldsymbol{q}}_i$ the pre-RoPE query. The kept keys preserve their relative order but occupy only p positions, so a T-token context consumes p positions of the window rather than T, which allows compacting more context than the original window holds. For night $i > 1$ the positions are offset by the slots of the frozen stack, and the slice is prefilled on top of that stack. On QuALITY (Qwen3.5-9B, 50×), the rank map fits the original attention output slightly better than keeping original positions (relative error 0.338 vs. 0.408) at a small cost in untrained accuracy (0.557 vs. 0.573), and training from it reaches 0.683 with ridge.

Regularizing value vectors with ridge regression. Re-rotation changes every query–key angle, so we refit $(\boldsymbol{\beta}, \boldsymbol{C}_v)$ to bridge the two position schemes. Each fit compares attention over the compact cache as it will be served, using the re-rotated keys and position-$(p+1)$ queries of Eq. (4), with attention over the original cache, using $\boldsymbol{K}$ and $\boldsymbol{q}_i$ at their true positions. We fit $\boldsymbol{\beta}$ so the served cache reproduces the original attention mass (by box-constrained least squares), and $\boldsymbol{C}_v$ so it reproduces the original attention output (by ridge regression):

$$\boldsymbol{\beta} = \log\Big(\arg\min_{\mathbf{w} \geq 0} \big\| \underbrace{\tilde{\boldsymbol{A}}\mathbf{w}}_{\text{served}} - \underbrace{\mathbf{m}}_{\text{original}} \big\|_2^2 \Big), \qquad \boldsymbol{C}_v = \arg\min_{\boldsymbol{C}} \big\| \underbrace{\tilde{\boldsymbol{X}}\boldsymbol{C}}_{\text{served}} - \underbrace{\boldsymbol{Y}}_{\text{original}} \big\|_F^2 + \lambda \|\tilde{\boldsymbol{X}}\|_2^2 \|\boldsymbol{C}\|_F^2, \tag{5}$$

where $\tilde{\boldsymbol{A}}_{ij} = \exp\big(\tilde{\boldsymbol{q}}_i (\tilde{\boldsymbol{C}}_k)_j^\top / \sqrt{d}\big)$ and row i of $\tilde{\boldsymbol{X}}$, $\mathrm{softmax}\big(\tilde{\boldsymbol{q}}_i \tilde{\boldsymbol{C}}_k^\top / \sqrt{d} + \boldsymbol{\beta}\big)$, attend the re-rotated keys, while $m_i = \sum_k \exp\big(\boldsymbol{q}_i \boldsymbol{K}_k^\top / \sqrt{d}\big)$ and row i of $\boldsymbol{Y}$, the original attention output for $\boldsymbol{q}_i$, attend the full cache. The fits force attention at the compressed positions to reproduce the attention the model computes at the original ones.

The ridge term stabilizes training from the initialization. Attention mass concentrates on few slots, so $\tilde{\boldsymbol{X}}$ is numerically rank-deficient even though the system is over-determined, and the unregularized value solve explodes along the poorly constrained directions, far past the norm of the original values (Appendix B.1.1). Training from that initialization collapses early, while ridge avoids the collapse

(Figure 3). Scaling the penalty by $\|\tilde{\boldsymbol{X}}\|_2^2$ makes λ a relative regularizer, and we use $\lambda = 10^{-4}$ for all heads and layers unless specified otherwise. On QuALITY, AM initialization with ridge regression beats the unregularized fit in every group and the original Cartridge initialization by $+0.131$ on average (Figure 9).

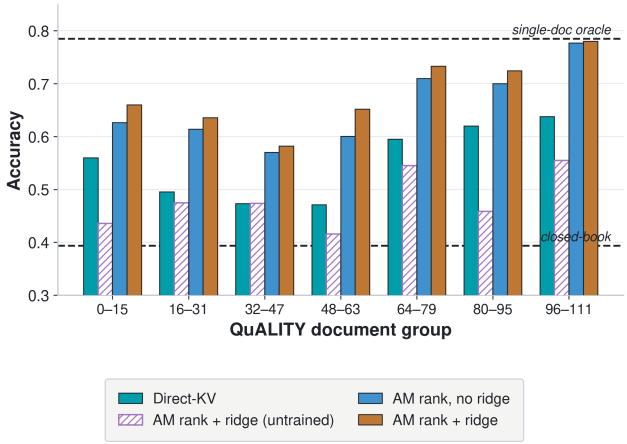


Figure 9: **Initialization and final accuracy on QuALITY** (Qwen3.5-9B, $20\times$, seven groups of 16 documents, each trained independently). AM initialization with ridge regression beats the unregularized fit in every group (mean 0.681 vs. 0.657) and the first-p-token (direct-KV) initialization by $+0.131$.

Model-specific details. On Llama-3.1, the 331-token system prompt is kept verbatim as the first slots, and all reference queries come from the slice itself. On Qwen3.5 and Gemma4, 80% of the reference queries come from self-study. Their recurrent and sliding-window layers hold no slots; the Cartridge stores the state those layers reach after reading the slice.

A.5 Baselines

Every baseline uses the same base model as Accordion to build and read its memory. We compare against the following methods.

- **ICL** keeps all past experience as raw text in the context window. On Harvey, the agent also keeps tool access to the full corpus.
- **ICL with summaries** keeps the newest content raw and replaces older content with summaries written by the same model once the window fills.
- **ACE** (Zhang et al., 2026) maintains a text playbook: a generator answers, a reflector extracts lessons, and a curator adds them to the playbook.
- **Dynamic Cheatsheet** (Suzgun et al., 2026) maintains a single text cheatsheet that the model rewrites as new material arrives.
- **LoRA** (Hu et al., 2022) stores memory in the weights: an adapter trained on the same self-study data as Accordion.
- **KVzip** (Kim et al., 2025) evicts KV entries by how well the context can be reconstructed from them, without training.
- **AM, no training** stacks the ridge AM initialization of Accordion (Appendix A.4) and never trains it.
- **Naive Cartridges** (Eyuboglu et al., 2026) trains each Cartridge alone, initialized from its first p tokens, and concatenates them at serving time.

We also report reference points that are not deployable baselines: a context ceiling, which places only the relevant documents in context; closed-book accuracy; and, on QuALITY, a joint-training bound that trains one Cartridge on all groups at once.

A.5.1 QuALITY

All QuALITY baselines use the same 11-group stream as ACCORDION and score every absorbed group after each stage. Memory budgets are matched to the $20\times$ Cartridge stack: 5,200 slots after g_0, 32,912 / 32,992 after g_6, and 52,496 / 52,592 after g_{10} (Qwen3.5-9B / Gemma4-12B-it). Text memories are capped at the same number of tokens, and the LoRA rank matches the bytes of the g_6 stack.

ICL with summaries. Older articles are replaced by same-model summaries ($50\times$) to fit a 262k-token window, leaving 41 raw and 135 summarized articles at 176.

ACE. We adapt ACE to fact recall. Self-study rows stream in chunks of 128. The generator answers from the playbook alone; the reflector sees that answer, the reference answer, and the article, and lists missing facts; the curator appends them to the article's section of the playbook. Each article has a quota of one sixteenth of the stage's budget increment and stops at its quota or after 16 rows without a new fact. Gemma curates at temperature 0 and Qwen at 0.7. Relative to the original, we drop the environment grade and the grow-and-refine step, curate in chunks, and cap the playbook. The published strategy prompts scored 0.433 on Gemma g_0, equal to closed book, and are not plotted.

Dynamic Cheatsheet. The original rewrites the cheatsheet after each query. We rewrite it once per group from 128 self-study rows (8 per article, 16 per curator call), compress it once if it exceeds the budget and then truncate, and freeze it while answering. Gemma curates greedily; Qwen curates at temperature 0.7 because its greedy curator looped.

LoRA. One adapter per chain on the attention and MLP projections (for Qwen, also the Gated-DeltaNet projections), with $\alpha = 2r$ and rsLoRA (Kalajdzievski, 2023): $r = 216$ for Qwen and $r = 66$ for Gemma, fixed through g_{10} ($\approx 63\%$ of the g_{10} stack). Stage k trains on the same packs as the chain, starting from stage $k-1$'s checkpoint. From a g_0 sweep, the learning rate is 3×10^{-5} for Qwen and 10^{-4} for Gemma.

Naive Cartridges and AM without training. Each naive Cartridge has the same size as the corresponding ACCORDION region and is trained for 2 epochs at learning rate 0.02 on its own group's rows. Gemma Cartridges carry a frozen sliding-window state; Qwen Cartridges carry no GatedDeltaNet state. The untrained AM stack builds each region conditioned on the previous stack. Gemma's AM stack drops from 0.433 to 0.345 at the boundary between dev and train articles; Qwen's does not.

Reference points. The single-group ceiling, closed book, and joint training are described in Appendix A.1.2.

A.5.2 CITIES

All CITIES baselines see the same fixed stream as ACCORDION (Appendix A.1.1). ICL keeps every experience in context. Untrained AM compacts the same slices into 512 slots per night, using original positions and no ridge. The naive-Cartridges baseline trains a 512-slot Cartridge alone on each slice, initialized from the key–value states of the slice's first tokens, and concatenates the Cartridges at serving time.

A.5.3 BLIND SPECTRUM MONITORING

All BSM baselines see the same 90 scans.

- **ICL** keeps every earlier scan verbatim in context (on Qwen3-32B the full trajectory exceeds the context window): 0.220 on Qwen3.5-9B and 0.223 on Qwen3-32B.
- **ICL, frontier models** gives Claude Opus 4.7 and Gemini 3.1 Pro the full trajectory: 0.373 and 0.367.
- **AM, no training** stacks the AM initialization of ACCORDION without training: 0.208 when compacting every 30 scans and 0.224 every 45, against 0.320 and 0.255 once trained.

- **Text summary of self-study data** serves a text summary of the same self-study rows, at the same cadence and compaction ratio: 0.28 on Qwen3-32B, against 0.345 for ACCORDION (Appendix B.3.2).

A.5.4 DATABASE SQLITE

All database baselines use Qwen3.8-27B and the same question ordering and acting settings. ICL retains the full trajectory. AM compacts without training at the same boundaries and target compression ratios. ACE replaces the preceding day's context with a text playbook, using its reflector and curator with a 4,000-token playbook cap. Table 9 reports all metrics from the same three trajectories for each baseline. Benchmark score is the mean of $\mathbf{1}[\text{correct}](1 - q/15)$, where q is the number of SQL queries used.

Table 9: **Database SQLite, 20 questions, Qwen3.8-27B.** Accuracy, benchmark score, and average SQL queries per question. Baselines average three trajectories; ACCORDION averages two continuations of the historical trajectory.

Method	Accuracy	Score	SQL queries
ICL	0.717	0.562	3.88
AM, no training	0.700	0.543	3.88
ACE (Zhang et al., 2026)	0.750	0.524	4.78
ACCORDION (ours)	**0.775**	**0.657**	**2.58**

A.5.5 HARVEY

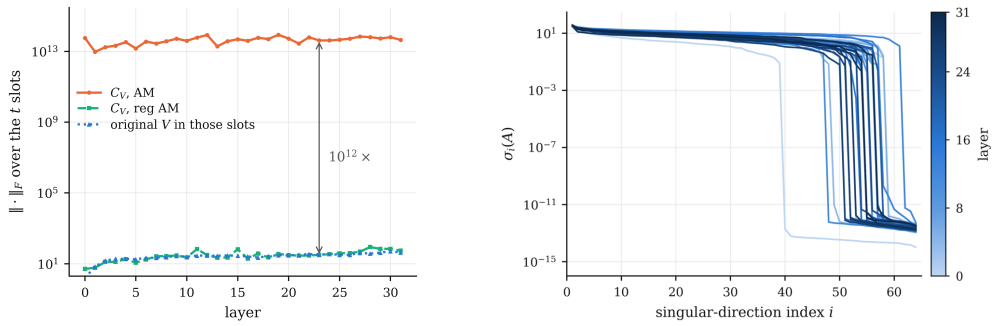
All Harvey baselines share the harness of Appendix A.1.5 and differ only in the memory attached at answer time.

- **Tools only** answers with tool access to the full store and no memory across answers. Within one answer its tool calls and results accumulate in context as usual, so it has in-episode ICL; nothing carries to the next answer.
- **KVzip** (Kim et al., 2025) prefills the same earlier transcripts that ACCORDION studies and keeps the KV entries with the highest attention-based scores, at $4\times$ compression and without training. The prefill is capped at 229,376 tokens on Qwen3.8-27B and 114,688 on Gemma4-31B-it.
- **ACE** (Zhang et al., 2026) runs a reflector and curator on the same model. They add lessons from the experience's trajectory to a playbook, capped at 4,000 tokens and then frozen, which is placed in the prompt for the next answer ($11.0\times$). Neither ACE nor ACCORDION sees rubric feedback; both learn from the transcript alone.
- **Oracle retrieval** places the documents of the matter under evaluation in context, up to 560K characters on Qwen3.8-27B and 280K on Gemma4-31B-it, and keeps tool access.

B ADDITIONAL RESULTS

B.1 ATTENTION MATCHING INITIALIZATION

B.1.1 RIDGE REGULARIZATION



(a) (b)

Figure 10: **Unregularized and ridge AM value solves** (one line per layer). (a) $\|C_v\|_F$ without ridge, with ridge, and for the original V of the selected tokens. (b) Singular values of the system matrix per layer. A Frobenius norm is dominated by a few slots and overstates a uniform blow-up.

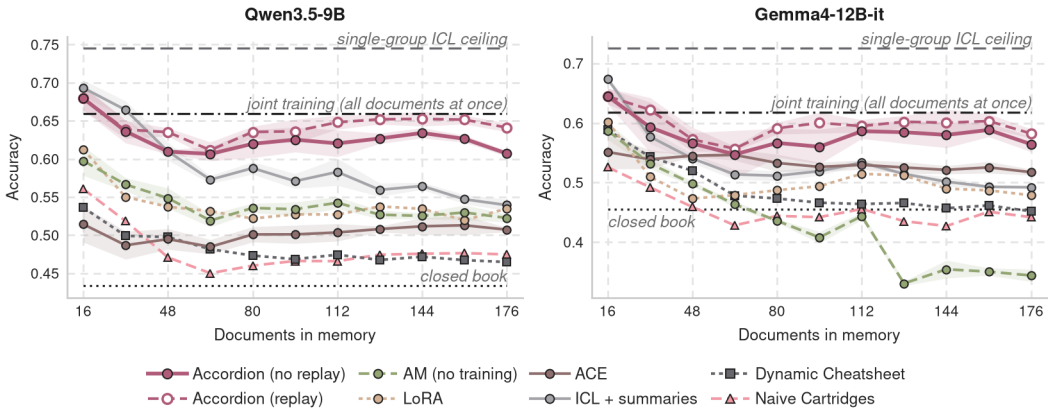


Figure 11: **QuALITY accuracy on seen documents with all baselines.** Accuracy on every document seen so far as 16 new documents are compacted each night. Bound lines mark the single-group ICL ceiling, closed book, and joint training on all documents at once.

B.2 QuALITY ablations

Figure 11 compares Accordion with every baseline on both models, and Figure 12 removes one component of Accordion at a time.

B.3 Blind Spectrum Monitoring ablations

Because the same model serves as teacher and student, any improvement on BSM has to come from the signal extracted during study rather than from a stronger teacher. We hypothesize that each Cartridge holds signal that emerges only with sufficient study and enough slots to store it. The ablations below test two parts of this: how often the agent compacts (Appendix B.3.1; Qwen3.5-9B) and whether the studied signal is kept better as latent KV or as text (Appendix B.3.2; Qwen3-32B).

B.3.1 Compaction frequency

Compacting less often degrades performance, and accuracy climbs as Cartridges are stacked, consistent with each round of study extracting additional signal. Each Cartridge has the same fixed slot count, so the schedule sets both how often the agent studies and how many scans share the same slots. Compacting every 15 scans ends the run with five Cartridges; every 30 or 45 scans ends it with two or one, each holding two or three times as many scans. Mean IoU falls from 0.358 to 0.320 and 0.255 (Figure 5).

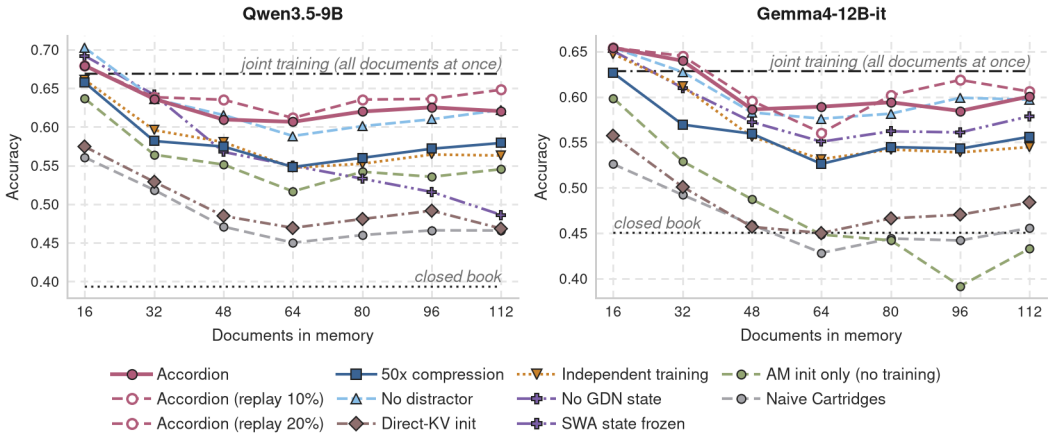


Figure 12: **QuALITY ablations of ACCORDION.** Accuracy on seen documents over the first seven nights (20× unless noted), removing one component of ACCORDION at a time.

B.3.2 LATENT KV VERSUS TEXT SUMMARY

Latent KV keeps more of the studied signal than text. We serve a recursive text summary of the same self-study rows, at the same cadence and compaction ratio as ACCORDION, in place of the Cartridge stack. On Qwen3-32B, the summary scores 0.28 against 0.345 for ACCORDION (Figure 13).

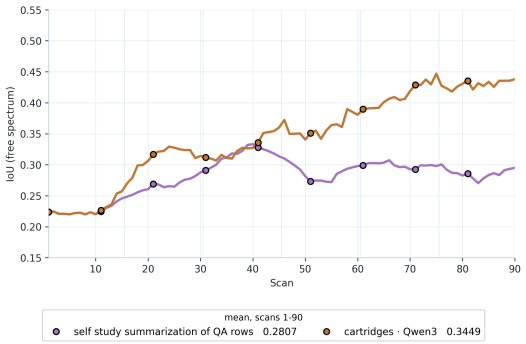


Figure 13: ACCORDION vs. a recursive text summary of the same self-study rows, at the same cadence and compaction ratio (Qwen3-32B, mean IoU, higher is better): 0.345 vs. 0.28.

B.4 DATABASE SQLITE ABLATIONS

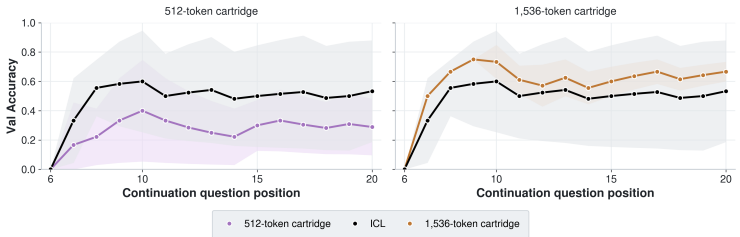


Figure 14: **Cartridge size on Database SQLite.** Validation accuracy over continuation questions 6–20 with 512-token and 1,536-token Cartridges, compared with ICL.

C Related work

We group prior work by where it stores experience. Throughout, our setting is sequential compaction of an agent's own trajectory into a growing stack of trained KV modules on a frozen model: each night, a new Cartridge is initialized with ridge-regularized Attention Matching, trained by stack-aware distillation on data dreamed from the consolidation slice, and then frozen.

Text-based summarization and textual memory. When the context fills, agents commonly summarize history into text and continue from the summary (Wang et al., 2025a; Packer et al., 2023). Generative Agents, A-MEM, and Mem0 store, link, and revise natural-language memory notes (Park et al., 2023; Xu et al., 2025; Chhikara et al., 2025). Dynamic Cheatsheet and Agentic Context Engineering (ACE) curate an evolving text memory or playbook across a task stream through generation, reflection, and incremental curation (Suzgun et al., 2026; Zhang et al., 2026); both are baselines in our experiments. Sleep-time compute also spends offline compute on the context before queries arrive, but its output is text (Lin et al., 2025b). Textual memories are inspectable, editable, and model-agnostic, which KV modules are not. The difference we study is the representation: our memory is a set of continuous key–value entries that the frozen model reads as context, optimized end-to-end against the model's own outputs instead of written as tokens.

Retrieval-based memory. Retrieval-augmented generation keeps an external store and fetches relevant items per query (Lewis et al., 2020); HippoRAG 2 extends this to associative, non-parametric continual learning (Gutiérrez et al., 2025). Retrieval can also select learned modules: Parametric RAG and a modular LoRA memory bank retrieve per-document adapters (Su et al., 2025; Zheng et al., 2026), and Cartridges at Scale (CAS) trains modular KV caches over a document collection and retrieves them at inference (Hardalov et al., 2026). CAS addresses interference through mixed-visibility training over relevant and distractor Cartridges, revisiting modules and examples from a shared collection, so modules can be adapted jointly. Our update policy trains only the newest Cartridge and keeps earlier ones unchanged but always in context. This is a difference in setting, not an empirical claim that CAS fails at sequential learning. Retrieval and internalization are compatible design choices, since trained Cartridges can themselves be retrieved.

KV-cache compression. Eviction methods retain a subset of cache entries by attention importance, attention sinks, or an observation window (Zhang et al., 2023; Xiao et al., 2024; Li et al., 2024); KVzip scores entries by context reconstruction so one compressed cache serves many queries (Kim et al., 2025) and is a baseline here. Attention Matching (AM) instead fits a compact cache in closed form to reproduce the attention output and mass of the original cache on reference queries (Zweiger et al., 2026). Its mass-preserving formulation targets compatibility with appended tokens, and it evaluates chunked and repeated compaction, which are precedents for composition and iteration; online AM and eviction for agent turns has since been studied with proxy queries (Liu et al., 2026). AM also tests L_2 regularization and reports a negative result (Zweiger et al., 2026, Appendix C.2); our positive result is specific to using the ridge solution as an initialization for training, and we do not claim to introduce regularization into AM for the first time. Learned compressors amortize compression in a trained network: prefix tuning, gist tokens, AutoCompressors, ICAE, Activation Beacon, KV-Distill, Compressed Context Memory, STILL, and Latent Context Compilation (Li & Liang, 2021; Mu et al., 2023; Chevalier et al., 2023; Ge et al., 2024; Zhang et al., 2025; Chari et al., 2025; Kim et al., 2024; O'Neill et al., 2026; Li et al., 2026). Cartridges trains a separate KV cache per corpus by self-study (Eyuboglu et al., 2026). Work on reusing independently computed caches repairs cross-chunk attention or trains caches to be concatenable (Yao et al., 2025; Du et al., 2026). Our method differs from these works in combining a per-slice cache optimized by gradient descent from an AM initialization, trained in the position it will be served (behind the frozen stack, ahead of a live window of varying length), and frozen afterwards. We do not compare against amortized compressors head-to-head.

Parameter updates. Fine-tuning on new data, including LoRA (Hu et al., 2022), faces the classical tension between acquisition and forgetting (Lopez-Paz & Ranzato, 2017), which also appears in sequential model editing (Gupta et al., 2024). MemoryLLM updates a dedicated memory pool (Wang et al., 2024), and sparse memory finetuning restricts updates to selected memory slots (Lin et al., 2025a). SEAL generates self-edits and applies persistent weight updates (Zweiger et al., 2025);

TTT-E2E, Titans, and Nested Learning build learners whose weights or memory update at test time (Tandon et al., 2025; Behrouz et al., 2025b;a), and Behrouz et al. (2026) consolidate context into memory in a sleep phase via parameter expansion and distillation. These methods change the learner or its weights, and several require new pretraining. Our backbone stays frozen; each stage allocates a new KV block and earlier blocks receive no gradient, so stored modules cannot be overwritten, although attention competition can still impair access to them. We include LoRA trained on the same dream data as a baseline.

Self-study and context distillation. Context distillation trains a student without a context to match a teacher that sees it (Snell et al., 2022). Cartridges' self-study generates synthetic conversations about corpus chunks and distills them into a KV cache (Eyuboglu et al., 2026); Active Reading generates study strategies and learning material from source text (Lin et al., 2026); synthetic continued pretraining (EntiGraph) diversifies data through an entity graph (Yang et al., 2025); Deep Context Distillation trains document LoRA knowledge modules with hidden-state and output supervision (Caccia et al., 2025); and generative adapters map a context to adapter weights in one forward pass (Chen et al., 2025). Recent analyses report that self-generated QA covers a document unevenly (Alimaskina et al., 2026) and that a distilled student can degrade when the internalized context reappears (Wang et al., 2026). These methods distill a fixed document or corpus, usually into a single module evaluated alone. We apply self-study to agent trajectories (task and action dreaming in the agent's own daytime format, with study views of the consolidation slice), and we distill under the conditioning the module will face at serving time: the frozen earlier stack, a live window of random length, and distractors.

D LIMITATIONS

Sleep-time cost. ACCORDION moves work from inference into training. Each compaction runs self-study and gradient-based distillation, which costs more than appending text or evicting KV entries. This cost is paid once per compaction and amortized over every later query, but it adds latency between an experience and its availability as memory, and it may not suit settings where experience must be usable immediately.

Bounded by the model's own study. Teacher and student are the same frozen model, so what a Cartridge can hold is bounded by what the model can extract from its experience during study. Where the model cannot recognize what is worth learning, compaction cannot recover it. We also require white-box access to the model's KV cache and gradients, which rules out closed models served only through an API.

Small but growing memory cost. Memory grows far more slowly than the raw context, but each compaction still adds a new cartridge, so it grows linearly with the number of compactions. Earlier cartridges are frozen and never revised, so outdated or incorrect knowledge persists once compacted. Over much longer horizons, to truly achieve indefinite ICL, the stack would need a way to merge, revise, or retire old cartridges. We leave this development open for future work.