RECURRENT HARNESS FOR AGENTIC CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Language model agents increasingly operate over long interaction streams, requiring online adaptation to distribution shifts through continual learning. However, existing agent harnesses typically either evolve their states only through forward information flow or rely on additional replay or validation data for harness optimization. To address this, we propose ReHarness, a recurrent harness framework for agentic continual learning that complements forward read and write with backward revision under a strictly online streaming protocol. We ask what information could have been captured differently in the harness state to better inform the current instance in the stream. Specifically, after each instance, we leverage the resulting trajectory as feedback to revise the reflections retrieved for that instance, thereby propagating feedback backward through the textual harness state. Across three continual learning benchmarks and three language model backbones, ReHarness consistently outperforms strong raw-context baselines, memory- and reflection-based methods, and prompt optimization approaches. Further analysis shows that ReHarness produces source-grounded and meaningful revisions, corrects overgeneralized or outdated experience under distribution shifts, and improves both stability and plasticity during continual learning.

1 INTRODUCTION

Language model agents (Guo et al., 2024; Wang et al., 2024a) increasingly move beyond isolated tasks to operate over long interaction streams (Ai et al., 2026; Asawa et al., 2026; He et al., 2026; Zhu et al., 2026). In these settings, agents must adapt to distribution shifts while retaining useful experience from earlier instances in those streams. Recent agent harnesses (Guo et al., 2026a; Ning et al., 2026) enable such continual learning (Wang et al., 2024b; Shi et al., 2025; Zheng et al., 2025) by maintaining persistent runtime scaffolding around a fixed large language model (LLM), often through memory (Chhikara et al., 2025; Zhang et al., 2025b; Dorovatas et al., 2026), reflection (Lingam et al., 2025; Yao et al., 2026; Zhang et al., 2026c), and prompts (Agrawal et al., 2026). This allows agents to improve online without repeatedly updating model parameters.

However, existing work typically evolves the harness state only through forward information flow. After each instance, these systems extract useful information (Chhikara et al., 2025; Yang et al., 2026; Zhang et al., 2026a) or reflect on the resulting trajectory (Lingam et al., 2025; Yao et al., 2026; Zhang et al., 2026c), integrate the experience into a stateful harness, and later retrieve from it (Wu et al., 2026b; Liu et al., 2026a; Zhang et al., 2026d) to guide future decisions. Nevertheless, once written, memories and reflections are rarely updated based on their effectiveness on later instances. While some methods (Chhikara et al., 2025; Suzgun et al., 2026; Zhang et al., 2026c) curate the harness state through operations such as add and delete, these operations primarily maintain a coherent memory or reflection bank rather than incorporate feedback from downstream use. Other methods (Cai et al., 2026; Kang et al., 2026; Lee et al., 2026; Zhang et al., 2026a) instead optimize the harness state using additional training or validation sets, which are unavailable under a strictly online streaming protocol. Under continual distribution shifts, outdated memories or overconfident reflections can therefore persist in the harness state and repeatedly misguide future decisions.

To address this gap, we propose ReHarness to close the recurrent loop for agentic continual learning. ReHarness complements the forward information flow in existing work with a backward revision mechanism. The revision asks what information from past experience could have been captured differently to better inform the current instance. Specifically, after each instance, we use a revision

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.

agent to revisit the reflections retrieved for that instance. We use the resulting trajectory as feedback for revision, while requiring any new content to remain grounded in the reflections' source trajectories. In this way, ReHarness propagates feedback backward through the textual harness state and carries the revised state forward to future instances. This enables online harness adaptation under a strictly streaming protocol without instance replay or gradient-based parameter updates. Together, forward read and write, coupled with backward revision, form a recurrent harness that can promptly correct outdated memories or overconfident reflections under distribution shifts.

We demonstrate that ReHarness consistently improves continual learning across three benchmarks (Ai et al., 2026; Asawa et al., 2026; He et al., 2026) and three backbone LLMs (Anthropic, 2026; Google DeepMind, 2026; OpenAI, 2026a). In particular, ReHarness outperforms strong raw-context baselines, structured memory and reflection methods (Chhikara et al., 2025; Lingam et al., 2025; Ouyang et al., 2026; Zhang et al., 2026c), and prompt optimization approaches (Agrawal et al., 2026). These improvements are statistically significant against all baselines, and the gains remain broadly consistent across the 16 evaluated tasks, including challenging settings where most baselines degrade relative to their stateless backbones. Further analysis confirms that ReHarness produces source-grounded, meaningful revisions and improves both stability and plasticity during continual learning. Qualitative examples and harness state dynamics illustrate how ReHarness corrects early overgeneralizations and revises outdated experience after distribution shifts. Finally, ablation studies show that ReHarness does not require a large revision set to outperform the strongest full-history baseline, while using roughly one-third as many tokens.

We summarize our contributions as follows:

- We propose ReHarness, which combines forward information flow with backward revision to form a recurrent harness for agentic continual learning in an online streaming setting.
- We show that ReHarness outperforms strong baselines across three continual learning benchmarks and three LLM backbones, with statistically significant gains consistent across challenging tasks.
- We demonstrate that ReHarness produces source-grounded and meaningful revisions, adapts overgeneralized or outdated reflections under distribution shifts, and improves stability and plasticity.

2 Related Work

Continual Learning. Continual learning (Wang et al., 2024b; Shi et al., 2025; Zheng et al., 2025) aims to adapt models to evolving data streams under distribution shifts while preventing catastrophic forgetting. For LLMs, recent work has extended this setting to test-time adaptation (Akyürek et al., 2025; Hu et al., 2025; Zuo et al., 2025; Feng et al., 2026) and self-evolving systems (Tao et al., 2024; Gao et al., 2026) that improve from interaction feedback. In agentic settings, learning signals are often accumulated through memory (Zhang et al., 2025b; Dorovatas et al., 2026; Luo et al., 2026), prompts (Agrawal et al., 2026), or persistent runtime states (Park et al., 2023; Karten et al., 2026), enabling online adaptation while keeping the underlying LLM parameters fixed. Recent benchmarks (Shridhar et al., 2021; Zhou et al., 2024; Ai et al., 2026; Asawa et al., 2026; Harrington et al., 2026; He et al., 2026) evaluate such systems over long interaction streams, measuring both plasticity to distribution shifts and stability of acquired capabilities. In this work, we study agentic continual learning under a strictly streaming protocol and improve online adaptation in memory space by revising accumulated experience based on its downstream effectiveness.

Agent Harness. Agent harnesses (Guo et al., 2026a; Ning et al., 2026) provide runtime scaffolding around a fixed LLM, turning text generation into goal-directed, multi-step behavior. A major line of work on agent harnesses for continual learning focuses on memory, including how raw experiences are stored (Packer et al., 2023; Chhikara et al., 2025; Wang et al., 2026; Rajesh et al., 2026), retrieved (Liu et al., 2026b; Guo et al., 2026b; Wu et al., 2026b; Liu et al., 2026a; Zhang et al., 2026d), and optimized (Kang et al., 2026; Karten et al., 2026; Lee et al., 2026; Li et al., 2026; Liu et al., 2026c; Ma et al., 2026; Pan et al., 2026; Xu et al., 2026; Zhang et al., 2026a;b) to support future instances. Some methods optimize memory offline (Cai et al., 2026; Ye et al., 2026), while others perform online in-context learning (Zheng et al., 2024; Sarukkai et al., 2025) by accumulating past demonstrations. More recent approaches instead abstract experience into high-level reflections (Lingam et al., 2025; Suzgun et al., 2026; Ouyang et al., 2026; Yao et al., 2026) and manage them through curation (Li & Huang, 2026; Wu et al., 2026a; Yan et al., 2026; Yang et al., 2026; Zhang et al., 2026c). However, existing approaches typically either rely on additional validation sets or trajectory replay for harness evolution, or support only forward information flow without

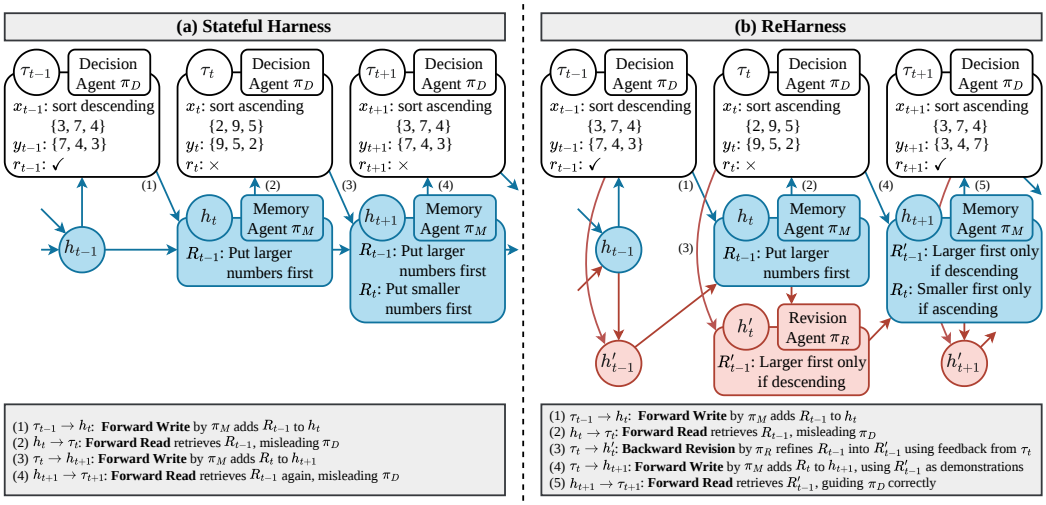

Figure 1: From stateful agent harnesses to ReHarness. (a) Traditional stateful harness propagates information through forward write and read, allowing an overgeneralized reflection to persist and mislead future instances. (b) ReHarness adds backward revision to refine retrieved reflections using downstream feedback, preventing the same error from propagating to future instances.

adapting previously learned experience based on its downstream effectiveness. Our work closes this recurrent loop in an online setting by complementing forward accumulation with backward revision.

3 Framework

We first motivate recurrent agent harnesses by identifying limitations of forward-only harness evolution in §3.1, then introduce textual backward feedback through revision to address the challenges of agentic continual learning in §3.2, and finally propose the ReHarness framework in §3.3.

3.1 From Stateful to Recurrent Agent Harnesses

An agent harness for continual learning (Ai et al., 2026; Asawa et al., 2026; Dorovatas et al., 2026; He et al., 2026) provides stateful runtime scaffolding around a decision agent π_D over a stream of instances. After observing the previous instance trajectory τ_{t-1}, a memory agent π_M produces a reflection R_{t-1} and writes it into the current harness state h_t. When the next instance arrives, the harness retrieves relevant reflections from h_t to assist π_D, which responds to query x_t with y_t and receives reward r_t from the environment. We denote the resulting trajectory of the current instance as $\tau_t = (x_t, y_t, r_t)$. This process repeats throughout the stream $\{\tau_t\}_{t=1}^{T}$. Existing work (Zhao et al., 2024; Lingam et al., 2025; Ouyang et al., 2026; Zhang et al., 2026c) thus evolves the harness state through alternating forward write ($\tau_{t-1} \to R_{t-1} \to h_t$) and forward read ($h_t \to R_{t-1} \to \tau_t$).

Figure 1 (a) illustrates this forward information flow in stateful harnesses with a simple sorting example. After correctly sorting $\{3, 7, 4\}$ in descending order at τ_{t-1}, π_M writes the reflection "put larger numbers first" into h_t. At τ_t, however, the query asks for ascending order. Retrieving R_{t-1} therefore misleads π_D, which produces an incorrect answer. Although π_M then writes a new reflection R_t recommending "put smaller numbers first," the misleading R_{t-1} remains in h_{t+1} and is retrieved again for τ_{t+1} because the queries contain overlapping numbers. Consequently, the same overgeneralized reflection misguides π_D again.

This example shows that a stateful harness with only the forward path is incomplete for continual learning. This forward-only reflection process does not account for the effectiveness of reflections when reused in future instances under distribution shifts. Misleading reflections can persist in the harness state and propagate errors through later retrievals. We instead propose a *recurrent* agent harness that combines forward information flow with backward feedback from future instances.

3.2 Textual Backward Feedback through Revision

Agentic continual learning under a strictly streaming setting poses unique challenges for backward feedback. First, each instance is observed only once, so we cannot return to past instances to col-

lect additional counterfactual outcomes for offline optimization (Cai et al., 2026; Li et al., 2026; Pan et al., 2026). Second, we do not have additional validation sets to select among candidate harness updates (Kang et al., 2026; Lee et al., 2026; Zhang et al., 2026b). Third, frontier API-based agents (Anthropic, 2026; Google DeepMind, 2026; OpenAI, 2026a) do not expose model parameters, precluding gradient-based updates. On the other hand, prior work (Pryzant et al., 2023; Shinn et al., 2023; Yuksekgonul et al., 2024) shows that natural language feedback can provide an effective learning signal without parameter updates. We therefore realize backward feedback directly in the textual harness state, allowing the harness to adapt online as the interaction stream evolves.

Specifically, we introduce a revision agent π_R that revisits reflections used for the current instance based on their observed effectiveness. Using the same sorting example in Figure 1 (b), after observing the current instance trajectory τ_t, π_R revises R_{t-1} in to R'_{t-1} using its source trajectory τ_{t-1} and the downstream evidence from τ_t. The revision aims to produce a reflection that, if used in place of R_{t-1}, could have prevented mistakes in τ_t. This typically involves incorporating previously missed evidence from τ_{t-1} or correcting an exaggerated hypothesis in R_{t-1}. Here, π_R adds a limiting condition to R'_{t-1} by specifying "larger first only if descending." Replacing R_{t-1} with R'_{t-1} yields a revised harness state h'_t, establishing the backward feedback path $\tau_t \to R'_{t-1} \to h'_t$. Meanwhile, we use R'_{t-1} as a demonstration for the forward write, which generates a new reflection R_t from τ_t with a similar limiting condition: "smaller first only if ascending." Combining the backward revision with this forward reflection yields the next harness state $h_{t+1} = h'_t \oplus R_t$, completing the recurrent update. Consequently, retrieving R'_{t-1} from h_{t+1} helps π_D answer τ_{t+1} correctly.

3.3 ReHarness: Closing the Recurrent Loop

We now instantiate this recurrent harness framework as ReHarness in Figure 1 (b).

Step 1: Forward Read. Let $h_t = \{R_i\}_{i<t}$ denote the current harness state. Given query x_t, we provide the decision agent π_D with the top-k relevant reflections retrieved from h_t:

$$\mathcal{I}_t = \{i_1, \ldots, i_k\} = \mathrm{Retrieval}(x_t, h_t, k), \qquad y_t = \pi_D(x_t, \{R_i\}_{i\in\mathcal{I}_t}). \tag{1}$$

The environment then provides reward r_t, forming the instance trajectory $\tau_t = (x_t, y_t, r_t)$.

Step 2: Backward Revision. After observing τ_t, the revision agent π_R independently revisits each retrieved reflection using its source trajectory τ_i and its effectiveness on the current instance τ_t:

$$R'_i = \pi_R(R_i, \tau_i; \tau_t), \qquad \forall i \in \mathcal{I}_t. \tag{2}$$

Importantly, τ_t serves only as feedback for assessing and revising R_i. Any new content in R'_i must remain grounded in its source trajectory τ_i to prevent information leakage. The revision may therefore incorporate evidence missed in τ_i, correct unsupported hypotheses in R_i, or refine the conditions under which the reflection should apply. Replacing the retrieved reflections with their revised versions while retaining all unretrieved reflections yields the intermediate state h'_t.

Step 3: Forward Write. Finally, the memory agent π_M reflects on τ_t to produce a new reflection:

$$R_t = \pi_M(\tau_t; \{R'_i\}_{i\in\mathcal{I}_t}), \qquad h_{t+1} = h'_t \oplus R_t. \tag{3}$$

We prompt π_M to explore operational evidence and retain only well-supported interpretations or strategies. The revised reflections $\{R'_i\}_{i\in\mathcal{I}_t}$ serve as demonstrations for π_M, while the prompt encourages R_t to capture complementary perspectives not covered by those demonstrations.

4 Experiment

4.1 Experiment Setup

Baselines. We compare ReHarness against the following strong baselines.

- **Mem0** (Chhikara et al., 2025) maintains atomic memories as harness state, retrieving top-k entries for the decision agent and incrementally extracting memories after each instance.
- **RBank** (Ouyang et al., 2026) maintains an append-only bank of reusable reflections as harness state and retrieves top-k reflections for each instance.
- **DoT** (Lingam et al., 2025) generates diverse reflections from the top-k retrieved prior experiences and uses them as the runtime harness for the current instance.
- **ACE** (Zhang et al., 2026c) maintains a global reflection playbook as its harness state and curates it through add, update, delete, and merge operations for coherence.

Table 1: Mean normalized gain across task streams for each benchmark–backbone combination. Highest gains are in bold. ReHarness outperforms all baselines, suggesting that backward revision of accumulated experience effectively adapts agent harnesses to continual distribution shifts.

Method	CL-Bench			MemoryBench			MemoryArena			Average
	Claude	GPT	Gemini	Claude	GPT	Gemini	Claude	GPT	Gemini	
Mem0	0.103	0.220	0.092	0.049	0.016	0.053	0.093	0.322	0.235	0.131
RBank	0.073	0.065	0.040	0.023	0.004	0.045	0.014	0.017	0.023	0.034
DoT	0.125	0.081	−0.008	0.053	0.027	0.010	0.129	0.179	0.042	0.071
ACE	0.042	0.088	0.036	0.085	−0.018	0.028	0.045	0.004	0.073	0.043
GEPA-Online	0.076	0.239	0.132	0.138	0.064	−0.134	0.126	0.129	0.058	0.092
Top-K Raw	0.165	0.268	0.190	0.035	0.041	0.037	0.145	0.340	0.291	0.168
Full Raw	0.191	0.299	**0.234**	0.063	0.024	−0.013	0.082	**0.358**	0.237	0.164
ReHarness	**0.217**	**0.317**	0.233	**0.160**	**0.096**	**0.064**	**0.305**	**0.358**	**0.408**	**0.240**

- **GEPA-Online** adapts GEPA (Agrawal et al., 2026) to our strictly online streaming setting, preserving reflective optimization of the decision prompt as persistent harness state, but replaces the original validation-based population search with sequential updates from each observed trajectory.
- **Full Raw** uses all prior instance trajectories in the stream as harness context, while **Top-K Raw** retrieves only the k most relevant trajectories for in-context learning.

We evaluate all methods with three frontier LLM backbones: Claude Opus 4.8 (Anthropic, 2026), GPT-5.5 (OpenAI, 2026a), and Gemini 3.7 Flash (Google DeepMind, 2026), using the same backbone for all agents within each method. We use each model's default generation configuration with high thinking effort and exclude hidden reasoning traces from all harness operations. Retrieval uses the top five entries ranked by Qwen3-Embedding-8B (Zhang et al., 2025a) and applies the same query-to-query comparison across all methods. We provide the prompts used by ReHarness in §A.

Dataset. We evaluate ReHarness against baselines on three continual learning benchmarks under a strictly online streaming protocol, with no training or validation set and no repeated trials.

- **CL-Bench** (Asawa et al., 2026) contains six tasks spanning cohort studies, codebase adaptation (Jimenez et al., 2024), database exploration (Hou et al., 2026), sales prediction, blind spectrum monitoring, and exploitable poker. Each task contains 12 to 120 multi-round instances with explicit distribution shifts and task-specific evaluation metrics.
- **MemoryBench** (Ai et al., 2026) contains long-input, long-output generation tasks across multiple domains. We use its five English tasks from HelloBench (Que et al., 2024) and WritingBench (Wu et al., 2026c), covering academic, creative, and political writing, with 17 to 47 single-round instances per task, evaluated by GPT-5.6-Sol (OpenAI, 2026b).
- **MemoryArena** (He et al., 2026) contains five domains spanning travel planning (Xie et al., 2024), web shopping (Yao et al., 2022), progressive web search (Chen et al., 2026), and formal reasoning in mathematics and physics, with 20 to 270 tasks per domain. Each task comprises 2 to 16 interdependent instances, each involving multiple interaction rounds.

Across these benchmarks, we standardize each task as a continual learning stream and each constituent interaction unit as an instance (i.e., τ_t). ReHarness and all baselines maintain harness state across instances within a stream and reset it between streams. To compare performance across tasks with different native metrics, we follow Asawa et al. (2026) and report normalized gain, $(r_{\text{method}} - r_{\text{stateless}})/(r_{\max} - r_{\text{stateless}})$, averaged over each stream. Here, $r_{\max}$ denotes the maximum achievable reward under each task's native metric. This metric quantifies the improvement over the stateless backbone agent enabled by learning from prior experience within the stream. We repeat each experiment over three independently sampled stream rollouts. Within each rollout, all methods share the same benchmark initialization and random seed, then evolve independently.

4.2 ReHarness Consistently Improves Agentic Continual Learning

Overall Performance. Table 1 reports the mean normalized gain across task streams for each benchmark–backbone combination. ReHarness achieves the highest overall normalized gain of 0.240 and ranks first or tied in eight of the nine combinations. Consistent with Asawa et al. (2026), raw-context methods are strong baselines: Top-K Raw and Full Raw outperform most structured memory and reflection methods on average, particularly on CL-Bench and MemoryArena. Despite

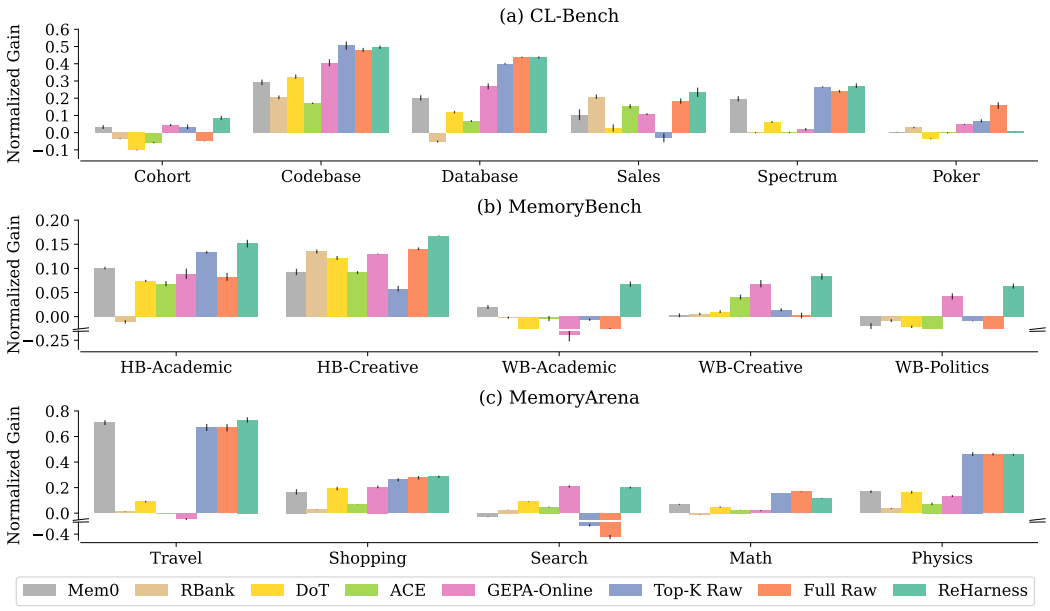

Figure 2: Task-level normalized gain for each method, averaged across the three backbone LLMs. Performance varies across tasks, while ReHarness remains effective on nearly all of them.

their strong performance, ReHarness is more token-efficient: on CL-Bench with Claude, Top-K Raw and Full Raw consume $1.3\times$ and $3.4\times$ as many total tokens, respectively, with detailed token and API cost breakdowns in §A. Reflection-based methods, such as RBank, DoT, and ACE, are less consistent, with some settings yielding negative gains. Similarly, GEPA-Online degrades substantially on MemoryBench with Gemini, suggesting that prompt optimization alone is less robust for continual learning. To assess statistical significance, we compare ReHarness against each baseline using two-sided paired Wilcoxon signed-rank tests with Bonferroni correction, treating each seeded task stream as one observation. ReHarness significantly outperforms all baselines ($p_{\text{adj}} < 0.05$), including the strong raw-context baselines Top-K Raw ($p_{\text{adj}} = 0.039$) and Full Raw ($p_{\text{adj}} = 0.043$). Overall, these results suggest that revising reflections with ReHarness is more effective than retaining raw history or relying on forward-only accumulation. The full raw results are in §A.

Task-Level Results. We next present the task-level breakdown for each benchmark in Figure 2, averaged across the three backbone LLMs. The gains vary substantially across tasks, with many methods showing marginal or even negative improvements over their stateless backbones. On CL-Bench, ReHarness leads or remains close to the best on most tasks, but shows only marginal gains on exploitable poker. On MemoryBench, while most methods perform well on HelloBench (HB-Academic and HB-Creative), only ReHarness achieves consistent gains across WritingBench (WB-Academic, WB-Creative, and WB-Politics). On MemoryArena, Mem0 is notably strong on travel planning, and GEPA-Online performs well on progressive web search, while ReHarness remains consistently competitive across all five domains. A closer examination suggests that, unlike the more deterministic codebase and database settings, exploitable poker is more stochastic: hand-level rewards can be weakly coupled with strategy quality due to hidden cards and chance. Raw-context baselines are less exposed to this credit-assignment issue because they preserve the original hand trajectories and defer their interpretation to the decision agent at each new hand. By contrast, ReHarness is less effective when downstream rewards provide a noisy signal for backward revision, and benefits most when feedback is diagnostic for revising reusable reflections.

5 Analysis

5.1 Is ReHarness Revision Source-Grounded?

We first examine whether backward revision in Equation 2 remains grounded in its source trajectory τ_i or imports information from τ_t. Following Min et al. (2023), we decompose each revised reflection R'_i into factual claims using GPT-5.6-Sol (OpenAI, 2026b). We classify each claim as: (1) grounded, if entailed by τ_i; (2) leaked, if not entailed by τ_i but entailed by τ_t; or (3) unsup-

Table 2: Average grounded, leakage, and unsupported claim rates of revised reflections on CL-Bench for each backbone, together with the average change per revision. The grounded rate remains above 95% for all three LLMs and decreases only marginally after revision, suggesting that ReHarness uses the downstream trajectory primarily as feedback rather than as a source of leaked information.

LLM	Grounded Rate (↑)		Leakage Rate (↓)		Unsupported Rate (↓)	
	Average	Δ / Revision	Average	Δ / Revision	Average	Δ / Revision
Claude	0.953	−0.003	0.021	0.006	0.026	0.002
GPT	0.987	−0.002	0.008	0.003	0.005	0.001
Gemini	0.963	0.007	0.013	0.004	0.024	−0.002

Table 3: Normalized gain under counterfactual revision on CL-Bench with Claude. Counterfactual replay reapplies the revised harness to the same instance and substantially improves over ReHarness and the no-revision baseline. This shows that the backward mechanism enables meaningful revision.

Revision Setting	Revision Based On	Revised Harness Applied To	CL-Bench						
			Cohort	Codebase	Database	Sales	Spectrum	Poker	Average
No Revision	–	–	−0.053	0.359	0.279	0.258	0.033	−0.012	0.144
ReHarness	τ_t	τ_{t+1}	−0.019	0.378	0.361	0.408	0.162	0.009	0.217
Counterfactual	τ_t	τ_t	**0.046**	**0.509**	**0.402**	**0.698**	**0.314**	**0.041**	**0.335**

ported, if entailed by neither. Table 2 reports the average rates across CL-Bench tasks for each backbone and their average change per revision. Across all backbones, over 95% of claims remain source-grounded, with leakage at 0.8–2.1% and unsupported claims at 0.5–2.6%. Moreover, revision changes these rates only marginally, with grounding decreasing by at most 0.3% per revision. These results indicate that the backbone LLMs largely follow the grounding constraint as intended, and ReHarness uses τ_t primarily as feedback rather than as a source of leaked information.

5.2 Is ReHarness Capable of Revision?

We next validate whether ReHarness revises the harness state as intended. The goal of revision is to use downstream feedback to improve previously learned experience for future instances. We therefore consider a counterfactual replay that modifies the online streaming protocol. After revising the harness state into h'_t based on the completed trajectory τ_t, we reuse h'_t to assist the decision agent on the same instance. This differs from ReHarness, which applies h'_t only to the next streamed instance τ_{t+1}. Importantly, h'_t remains grounded in past trajectories and uses τ_t only as feedback on their effectiveness. The replay thus asks: given the current outcome, what information already available in past trajectories should have been captured differently to better inform that instance?

Table 3 reports task-level normalized gains on CL-Bench with Claude for ReHarness, counterfactual replay, and a no-revision baseline that removes the revision agent π_R and retains only forward information flow. Counterfactual replay substantially improves over ReHarness across all tasks. This suggests that revision can recover or correct information from past trajectories using current feedback, producing a more effective harness state had those revisions been available beforehand. Hence, counterfactual replay confirms that the backward mechanism produces meaningful revisions. However, it violates the online streaming protocol. ReHarness instead applies the revised state h'_t only to the next instance τ_{t+1}, using feedback from τ_t to improve future instances. Compared with the no-revision baseline, ReHarness achieves higher gains across all tasks, especially in database exploration, sales prediction, and blind spectrum monitoring. This indicates that backward revision remains beneficial when applied only to subsequent instances in the stream.

5.3 What Does ReHarness Revise during Continual Learning?

Now that we have established the source grounding and capability of ReHarness revision, we use qualitative examples from CL-Bench with Claude in Figure 3 to examine what it revises. Figure 3 (a) shows an early-stage example from blind spectrum monitoring, where the agent infers persistent transmitters from sequential spectrum scans. After observing τ_2, the memory agent π_M overgeneralizes the evidence and recommends "report[ing] plausible additional channels to fill large silent gaps." Because the task rewards overlap with the ground-truth occupied spectrum, both missing real transmitters and hallucinating nonexistent ones hurt performance. When this reflection is retrieved from h_3 for τ_3, the decision agent π_D follows the speculative guidance, adds unobserved transmit-

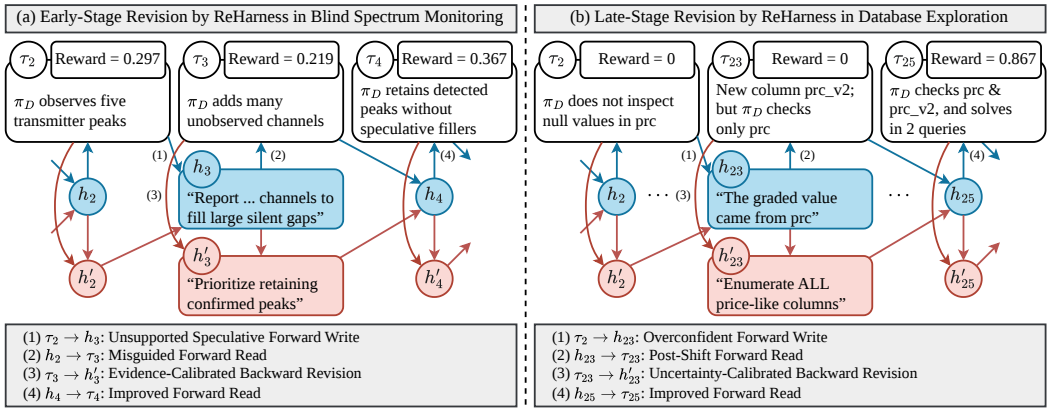

Figure 3: Qualitative examples of ReHarness revision during continual learning. (a) In blind spectrum monitoring, ReHarness corrects an early overgeneralized reflection with evidence-grounded guidance. (b) In database exploration, ReHarness adapts an outdated reflection to a schema shift.

Table 4: Stability–plasticity decomposition on CL-Bench. ReHarness achieves the highest average gains in both, showing robust transfer across distribution shifts and effective adaptation afterward.

Method	Claude		GPT		Gemini		Average	
	Stability	Plasticity	Stability	Plasticity	Stability	Plasticity	Stability	Plasticity
Mem0	0.002	0.101	0.014	0.206	−0.026	0.118	−0.003	0.142
RBank	**0.004**	0.069	0.001	0.065	−0.022	0.062	−0.006	0.065
DoT	−0.003	0.127	−0.002	0.083	0.000	−0.007	−0.002	0.068
ACE	0.000	0.042	0.011	0.077	−0.028	0.065	−0.006	0.061
GEPA-Online	0.000	0.076	0.006	0.223	−0.017	0.149	−0.004	0.153
Top-K Raw	−0.012	0.177	0.009	0.260	0.011	0.179	0.003	0.205
Full Raw	−0.005	0.196	0.010	0.290	**0.024**	0.210	0.010	0.232
ReHarness	0.003	**0.214**	**0.025**	**0.291**	0.016	**0.217**	**0.015**	**0.241**

ters, and receives a lower reward. Using this outcome as backward feedback, the revision agent π_R marks the hypothesis as "speculative" and changes the guidance to "prioritize retaining confirmed peaks over adding unconfirmed ones," producing h'_3. When the revised reflection is retrieved from h_4, π_D avoids speculative fillers and substantially improves the reward. This example shows how ReHarness corrects an early overgeneralization before it misguides future decisions.

The second example in Figure 3 (b) shows a late-stage revision from database exploration in CL-Bench, where the agent queries an unknown SQLite database. During τ_2, π_D fails to inspect null values in a price column "prc," yet π_M eventually consolidates the reflection that "the graded value came from prc" after repeated use in the early stream. After τ_{20}, CL-Bench introduces a distribution shift by changing the schema and adding "prc_v2." When the old-schema reflection is retrieved from h_{23} for τ_{23}, π_D checks only "prc," while the correct answer lies in "prc_v2." Using this failure as backward feedback, π_R changes the rule to "not assume which column is graded" and "enumerate all price-like columns," producing h'_{23}. When this revised reflection is retrieved for τ_{25}, π_D checks both "prc" and "prc_v2" and obtains the correct answer in two queries, substantially improving the reward. This example shows how ReHarness revises outdated strategies under distribution shifts through backward feedback. We provide the full trajectories, reflections, and revisions in §A.

5.4 How Does ReHarness Revise during Continual Learning?

We then investigate how ReHarness revises accumulated experience during continual learning. Following Asawa et al. (2026), we decompose normalized gain on CL-Bench into stability and plasticity. Stability measures gain on the first instance after a distribution shift, before post-shift feedback is available. Plasticity measures gain on subsequent instances before the next shift, where methods can adapt to the current distribution. Because the two categories contain different numbers of instances, their absolute gains are not directly comparable. As shown in Table 4, Mem0, RBank, DoT, ACE, and GEPA-Online exhibit negative stability gains on average, indicating that their forward-only harness states can become detrimental under distribution shifts. In contrast, ReHarness achieves the highest stability gain on average and the highest plasticity gain for every backbone LLM. These

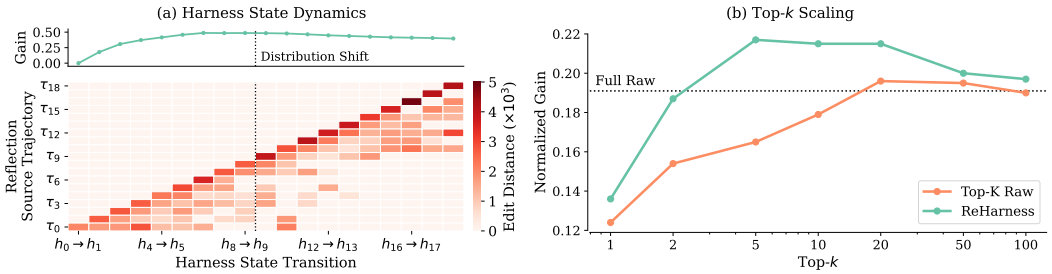

Figure 4: (a) Harness state dynamics by ReHarness on CL-Bench codebase adaptation with Claude. Revisions diminish as reflections stabilize before the distribution shift, then rise afterward to update outdated reflections. (b) Top-k scaling on CL-Bench with Claude. ReHarness improves sharply up to $k = 5$ and stabilizes for moderate k, while larger values yield diminishing returns and higher cost.

results suggest that backward revision enables more robust transfer across shifts while supporting effective adaptation once post-shift feedback becomes available, as illustrated in Figure 3.

We further visualize harness state dynamics on CL-Bench codebase adaptation with Claude. Each transition from h_t to h_{t+1} adds a new reflection R_t through forward write and revises retrieved reflections $\{R'_i\}_{i \in \mathcal{I}_t}$ through backward revision. In Figure 4 (a), diagonal heatmap entries mark new reflections, while later non-zero entries in the same row indicate revisions. We measure revision magnitude by character-level edit distance between the old and revised reflections from the same source trajectory. Before the distribution shift at τ_9, revisions to older reflections diminish as the harness state stabilizes and the gain rises. After the shift, older reflections require larger revisions to adapt to the new distribution. For example, the transition from h_{10} to h_{11} substantially revises reflections from τ_0 and τ_1. These revisions correct outdated experience while the gain remains relatively stable after the shift. As the stream progresses, revision increasingly concentrates on recent reflections relevant to the shifted distribution. Together, these dynamics show how ReHarness balances stability and plasticity by preserving useful experience while revising outdated reflections.

5.5 What Makes ReHarness Effective?

Finally, we conduct ablations to identify what makes ReHarness effective. In Figure 4 (b), we vary the number of retrieved and subsequently revised reflections, k, on a log scale for ReHarness and Top-K Raw on CL-Bench with Claude. Top-K Raw approaches Full Raw as k reaches 20, suggesting diminishing returns from additional in-context examples. For ReHarness, normalized gain rises sharply from $k = 1$ to 5, remains relatively stable for $k \in \{5, 10, 20\}$, and declines for larger k. This suggests that overly large k can introduce unnecessary revision and potential reflection drift while increasing computational cost. ReHarness therefore does not require a large revision set. We fix $k = 5$ throughout the main experiments and report this sweep as a post-hoc sensitivity analysis.

Furthermore, we ablate two components of ReHarness. The effect of removing backward revision is already established in Table 3. Removing the revised reflections $\{R'_i\}_{i \in \mathcal{I}_t}$ used as demonstrations for π_M in Equation 3 reduces the normalized gain on CL-Bench with Claude from 0.217 to 0.191, indicating that revised experience helps guide subsequent reflection generation.

6 Discussion

In this paper, we propose ReHarness for agentic continual learning beyond forward-only harness evolution. ReHarness complements forward read and write with backward revision, using downstream feedback to revise past reflections while grounding new content in their source trajectories. Across three continual learning benchmarks and three LLM backbones, ReHarness consistently outperforms strong baselines, with statistically significant gains across challenging tasks. Further analyses show that ReHarness produces source-grounded and meaningful revisions, corrects over-generalized or outdated reflections under distribution shifts, and improves stability and plasticity. Together, our results highlight recurrent harness adaptation as a promising direction for continually refining accumulated experience under a strictly streaming setting. Despite these gains, ReHarness underperforms on exploitable poker in CL-Bench, where downstream rewards provide noisy signals for backward revision. Moreover, while ReHarness is more token-efficient than the strongest full-history raw-context baseline, it still stores raw trajectories for future revision, incurring additional storage overhead. We aim to address these limitations in future work.

AI USE STATEMENT

We used LLMs to edit and polish the manuscript for clarity and presentation, and AI coding tools to assist with code implementation for our experiments. The first author reviewed and verified all AI-assisted code and experiment results before including them in this work. We take full responsibility for the final content of the paper, including all text, claims, code, and experiment results produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the full ReHarness framework in §3, the prompts in §A, and the benchmarks, baselines, model backbones, retrieval settings, and evaluation protocol in §4.1. Together, these materials provide the implementation and evaluation details needed to reproduce our experiments.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *ICLR*, 2026.

Qingyao Ai, Yichen Tang, Changyue Wang, Jianming Long, Weihang Su, and Yiqun LIU. Memorybench: A benchmark for memory and continual learning in LLM systems. In *ICML*, 2026.

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. In *ICML*, 2025.

Anthropic. Claude opus 4.8 system card. `https://www.anthropic.com/claude-opus-4-8-system-card`, 2026. Accessed: 2026-08-15.

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*, 2026.

Yishuo Cai, Xingyu Guo, Xuancheng Huang, Jinhua Du, can huang, Wenxuan Huang, Wenhan Ma, Yuyang Hu, Aohan Zeng, Jie Tang, and Xu Sun. From player to master: Enhancing test-time learning of LLM agents via reinforcement learning over memory. In *ICML*, 2026.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Sahel Sharifymoghaddam, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, et al. Browsecomp-plus: a fair and disentangled evaluation benchmark for deep search agents. In *ACL*, 2026.

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.

Vaggelis Dorovatas, Malte Schwerin, Andrew D. Bagdanov, Lucas Caccia, Antonio Carta, Laurent Charlin, Barbara Hammer, Tyler L. Hayes, Timm Hess, Christopher Kanan, Dhireesha Kudithipudi, Xialei Liu, Vincenzo Lomonaco, Jorge Mendez-Mendez, Darshan Patil, Ameya Prabhu, Elisa Ricci, Tinne Tuytelaars, Gido M van de Ven, Liyuan Wang, Joost van de Weijer, Jonghyun Choi, Martin Mundt, and Rahaf Aljundi. Position: Modular memory is the key to continual learning agents. In *ICML Position Paper Track*, 2026.

Guhao Feng, Shengjie Luo, Kai Hua, Ge Zhang, Wenhao Huang, Di He, and Tianle Cai. In-place test-time training. In *ICLR*, 2026.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Qihan Ren, Yiran Wu, Hongru WANG, Han Xiao, Yuhang Zhou, Shaokun Zhang, Jiayi Zhang, Jinyu Xiang, Yixiong Fang, Qiwen Zhao, Dongrui Liu, Cheng Qian, Zhenhailong Wang, Minda Hu, Huazheng Wang, Qingyun Wu, Heng Ji, and Mengdi Wang.

A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *Transactions on Machine Learning Research*, 2026.

Google DeepMind. Gemini 3.7 flash: Model card. `https://deepmind.google/models/model-cards/gemini-3-7-flash/`, 2026. Accessed: 2026-08-15.

Jianyuan Guo, Zhiwei Hao, Chengcheng Wang, Cheng Fan, Tingzhang Luo, Hongguang Li, Ying Gao, Hefei Mei, Jiankun Peng, Rongjian Xu, et al. From question answering to task completion: A survey on agent system and harness design. *arXiv preprint*, 2026a.

Siyuan Guo, Yali Du, Hechang Chen, Yi Chang, and Jun Wang. Cascade: Case-based continual adaptation for large language models during deployment. *arXiv preprint*, 2026b.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: a survey of progress and challenges. In *IJCAI*, 2024.

Anne Harrington, Nayan Saxena, Michael Murphy, Anastasia Borovykh, Zeyu Yun, Sridhar Kamath, Ara Eindra Kyi, Trevor Darrell, Jitendra Malik, and Yutong Bai. When does continual learning require learning. *arXiv preprint*, 2026.

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, Jiaxin Pei, Julian McAuley, Yejin Choi, and Alex Pentland. Benchmarking agent memory in interdependent multi-session agentic tasks. In *ICML*, 2026.

Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation: Benchmarking llms as semantic encoders. In *ACL*, 2026.

Jinwu Hu, Zitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-time learning for large language models. In *ICML*, 2025.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *ICLR*, 2024.

Borui Kang, Jinrui Gu, Junhan Lv, Wenbin Li, Lei Wang, and Yang Gao. Harness continual learning: Continual adaptation beyond model parameters. *arXiv preprint*, 2026.

Seth Karten, Joel Zhang, Tersoo Upaa Jr, Ruirong Feng, Wenzhe Li, Chengshuai Shi, Chi Jin, and Kiran Vodrahalli. Continual harness: Online adaptation for self-improving foundation agents. *arXiv preprint*, 2026.

Yoonho Lee, Roshen Sanjay Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-harness: End-to-end optimization of model harnesses. In *COLM*, 2026.

Jian Li and Hua Huang. Textual stochastic gradient descent: Discrete optimization of external memory for reasoning language agents. In *ICML*, 2026.

Yibo Li, Zijie Lin, Ailin Deng, Xuan Zhang, Yufei He, Shuo Ji, Tri Cao, and Bryan Hooi. Just-in-time reinforcement learning: Continual learning in LLM agents without gradient updates. In *ICML*, 2026.

Vijay Lingam, Behrooz Omidvar Tehrani, Sujay Sanghavi, Gaurav Gupta, Sayan Ghosh, Linbo Liu, Jun Huan, and Anoop Deoras. Enhancing language model agents using diversity of thoughts. In *ICLR*, 2025.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Simplemem: Efficient lifelong memory for LLM agents. In *ICML*, 2026a.

Jiaqi Liu, Xinyu Ye, Peng Xia, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Evolvemem: Self-evolving memory architecture via autoresearch for llm agents. *arXiv preprint*, 2026b.

Zewen Liu, Zhan Shi, Yisi Sang, Bing He, Minhua Lin, Tianxin Wei, Dakuo Wang, Benoit Dumoulin, Wei Jin, and Hanqing Lu. Adaptive auto-harness: Sustained self-improvement for agentic system deployment on open-ended task streams. *arXiv preprint*, 2026c.

Jinghao Luo, Yuchen Tian, Chuxue Cao, Ziyang Luo, Hongzhan Lin, Kaixin Li, Chuyi Kong, Ruichao Yang, and Jing Ma. From storage to experience: A survey on the evolution of llm agent memory mechanisms. In *ACL Findings*, 2026.

Yiwen Ma, Songjun Tu, Qichao Zhang, Dong Li, Linjing Li, and Dongbin Zhao. Memchain: Learning interpretable memory traces for memory-augmented llm agents. *arXiv preprint*, 2026.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *EMNLP*, 2023.

Xuying Ning, Katherine Tieu, Dongqi Fu, Tianxin Wei, Zihao Li, Yuanchen Bei, Jiaru Zou, Mengting Ai, Zhining Liu, Ting-Wei Li, et al. Code as agent harness. *arXiv preprint*, 2026.

OpenAI. GPT-5.5 system card. `https://openai.com/index/gpt-5-5-system-card/`, 2026a. Accessed: 2026-08-15.

OpenAI. GPT-5.6 system card. `https://deploymentsafety.openai.com/gpt-5-6`, 2026b. Accessed: 2026-08-15.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long 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 *ICLR*, 2026.

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*, 2023.

Wenbo Pan, Shujie Liu, Chin-Yew Lin, Jingying Zeng, Xianfeng Tang, Xiangyang Zhou, Yan Lu, and Xiaohua Jia. Retrospective harness optimization: Improving llm agents via self-preference over trajectory rollouts. *arXiv preprint*, 2026.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *UIST*, 2023.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *EMNLP*, 2023.

Haoran Que, Feiyu Duan, Liqun He, Yutao Mou, Wangchunshu Zhou, Jiaheng Liu, Wenge Rong, Zekun Moore Wang, Jian Yang, Ge Zhang, et al. Hellobench: Evaluating long text generation capabilities of large language models. *arXiv preprint*, 2024.

Shreyas Rajesh, Pavan S Holur, Mehmet Yigit Turali, Chenda Duan, and Vwani Roychowdhury. Panini: Continual learning in token space via structured memory. In *ICML*, 2026.

Vishnu Sarukkai, Zhiqiang Xie, and Kayvon Fatahalian. Self-generated in-context examples improve LLM agents for sequential decision-making tasks. In *NeurIPS*, 2025.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. {ALFW}orld: Aligning text and embodied environments for interactive learning. In *ICLR*, 2021.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. In *EACL*, 2026.

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. A survey on self-evolution of large language models. *arXiv preprint*, 2024.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of computer science*, 2024a.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE TPAMI*, 2024b.

Ziyi Wang, Haonan Jin, Zian Wang, Wendong Wang, and Lanshan Zhang. LightWM: Training-free hierarchical working memory for small language model agents. In *ICML*, 2026.

Beining Wu, Zihao Ding, Jun Huang, and Yanxiao Zhao. Forget to improve: On-device llm-agent continual learning via budget-curated memory. *arXiv preprint*, 2026a.

Yifan Wu, Lizhu Zhang, Yuhang Zhou, Mingyi Wang, Bo Peng, Serena Li, Xiangjun Fan, and Zhuokai Zhao. Remember when it matters: Proactive memory agent for long-horizon agents. *arXiv preprint*, 2026b.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Wang Zijia, Ji Zhang, Mengyue Wu, Qin Jin, and Fei Huang. Writingbench: A comprehensive benchmark for generative writing. In *NeurIPS Datasets and Benchmarks Track*, 2026c.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. Travelplanner: A benchmark for real-world planning with language agents. In *ICML*, 2024.

Weijia Xu, Alessandro Sordoni, Chandan Singh, Zelalem Gero, Michel Galley, Xingdi Yuan, and Jianfeng Gao. Test-time learning with an evolving library. *arXiv preprint*, 2026.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, et al. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *ACL*, 2026.

Ke Yang, Zixi Chen, Xuan He, Jize Jiang, Michel Galley, Chenglong Wang, Jianfeng Gao, Jiawei Han, and ChengXiang Zhai. Plugmem: A task-agnostic plugin memory module for LLM agents. In *ICML*, 2026.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In *NeurIPS Datasets and Benchmarks Track*, 2022.

Tianjun Yao, Yongqiang Chen, Yujia Zheng, Pan Li, Zhiqiang Shen, and Kun Zhang. Parammem: Augmenting language agents with parametric reflective memory. In *ICML*, 2026.

Yongshi Ye, Hui Jiang, Feihu Jiang, Tian Lan, Yichao Du, Biao Fu, Xiaodong Shi, Qianghuai Jia, Longyue Wang, and Weihua Luo. UMEM: Unified memory extraction and management framework for generalizable memory. In *ICML*, 2026.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic" differentiation" via text. *arXiv preprint*, 2024.

Guibin Zhang, Haotian Ren, Chong Zhan, Junhao Wang, He Zhu, Wangchunshu Zhou, and Shuicheng YAN. Memevolve: Meta-evolution of agent memory systems. In *ICML*, 2026a.

Hangfan Zhang, Shao Zhang, Kangcong Li, Chen Zhang, Yang Chen, Yiqun Zhang, Lei Bai, and Shuyue Hu. Self-harness: Harnesses that improve themselves. *arXiv preprint*, 2026b.

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 *ICLR*, 2026c.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint*, 2025a.

Yunxiang Zhang, Yiheng Li, Ali Payani, and Lu Wang. AdaMEM: Test-time adaptive memory for language agents. In *ICML*, 2026d.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 2025b.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *AAAI*, 2024.

Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. Towards lifelong learning of large language models: A survey. *ACM Computing Surveys*, 2025.

Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. Synapse: Trajectory-as-exemplar prompting with memory for computer control. In *ICLR*, 2024.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *ICLR*, 2024.

Deyao Zhu, Xin Zhou, Shengling Qin, Xuekai Zhu, Hangliang Ding, Shu Zhong, Zixin Wen, Zhonglin Xie, Chenhui Gou, Linxuan Ren, et al. Edgebench: Unveiling scaling laws of learning from real-world environments. *arXiv preprint*, 2026.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-time reinforcement learning. In *NeurIPS*, 2025.

A APPENDIX

We report task-level normalized gains for each LLM on CL-Bench, MemoryBench, and MemoryArena in Tables 5, 6, and 7, respectively. We provide the prompts used by the decision agent π_D, memory agent π_M, and revision agent π_R in ReHarness in Figure 5. We also provide the full qualitative revision examples discussed in §5.3 for blind spectrum monitoring and database exploration in Figures 6 and 7, respectively.

We further examine the token consumption and API cost of ReHarness. Despite requiring one reflection call for forward write and k revision calls for backward revision, in addition to the decision calls shared by all methods, ReHarness consumes fewer total tokens than the two strong raw-context baselines. This is because compact reflections substantially reduce the repeated decision-time context cost of raw trajectories, especially for long, multi-round instances. Across one rollout over all CL-Bench tasks with Claude, ReHarness consumes $\sim$37.19M input and $\sim$4.38M output tokens, totaling $\sim$41.57M. By comparison, Top-K Raw consumes $\sim$52.20M input and $\sim$0.46M output tokens ($\sim$52.66M total), while Full Raw consumes $\sim$141.68M input and $\sim$0.44M output tokens ($\sim$142.12M total), or $1.3\times$ and $3.4\times$ as many total tokens as ReHarness, respectively. Thus, although ReHarness generates more output tokens, its input-token savings more than offset this overhead in total consumption. With prompt caching and Claude API pricing, the corresponding costs are approximately \$85 for Top-K Raw, \$254 for ReHarness, and \$314 for Full Raw. ReHarness therefore remains substantially cheaper than Full Raw while achieving a higher normalized gain (0.217 vs. 0.191), whereas Top-K Raw is cheaper but achieves a lower gain (0.165).

Table 5: Normalized gain for each LLM on each CL-Bench task, with the highest value in bold.

Method	Cohort	Codebase	Database	Sales	Spectrum	Poker	Average
LLM: Claude Opus 4.8							
Mem0	-0.045 ± 0.014	0.408 ± 0.022	-0.016 ± 0.014	0.284 ± 0.011	0.000 ± 0.018	-0.013 ± 0.002	0.103 ± 0.006
RBank	-0.065 ± 0.002	0.238 ± 0.007	-0.119 ± 0.006	0.376 ± 0.011	0.000 ± 0.001	0.005 ± 0.003	0.073 ± 0.002
DoT	-0.074 ± 0.002	$\mathbf{0.499 \pm 0.009}$	0.056 ± 0.006	0.252 ± 0.026	0.033 ± 0.004	-0.017 ± 0.001	0.125 ± 0.007
ACE	$\mathbf{-0.015 \pm 0.004}$	0.231 ± 0.004	-0.005 ± 0.004	0.051 ± 0.006	0.000 ± 0.001	-0.010 ± 0.003	0.042 ± 0.001
GEPA-Online	-0.028 ± 0.006	0.238 ± 0.023	0.040 ± 0.014	0.171 ± 0.004	-0.023 ± 0.004	$\mathbf{0.056 \pm 0.002}$	0.076 ± 0.008
Top-K Raw	-0.119 ± 0.010	0.179 ± 0.006	0.371 ± 0.005	0.267 ± 0.020	$\mathbf{0.275 \pm 0.003}$	0.014 ± 0.005	0.165 ± 0.003
Full Raw	-0.047 ± 0.002	0.364 ± 0.010	$\mathbf{0.406 \pm 0.004}$	0.229 ± 0.013	0.216 ± 0.007	-0.020 ± 0.012	0.191 ± 0.004
ReHarness	-0.019 ± 0.012	0.378 ± 0.011	0.361 ± 0.004	$\mathbf{0.408 \pm 0.018}$	0.162 ± 0.014	0.009 ± 0.002	$\mathbf{0.217 \pm 0.006}$
LLM: GPT-5.5							
Mem0	-0.047 ± 0.006	0.402 ± 0.019	0.349 ± 0.012	0.365 ± 0.033	0.233 ± 0.014	0.020 ± 0.002	0.220 ± 0.006
RBank	-0.029 ± 0.003	0.319 ± 0.009	-0.053 ± 0.005	0.068 ± 0.005	0.000 ± 0.001	0.087 ± 0.004	0.065 ± 0.002
DoT	-0.105 ± 0.002	0.228 ± 0.010	0.226 ± 0.005	0.078 ± 0.016	0.138 ± 0.002	-0.080 ± 0.005	0.081 ± 0.005
ACE	-0.109 ± 0.003	0.152 ± 0.001	0.118 ± 0.006	0.322 ± 0.012	0.000 ± 0.001	0.044 ± 0.004	0.088 ± 0.003
GEPA-Online	0.121 ± 0.006	$\mathbf{0.694 \pm 0.014}$	0.307 ± 0.011	0.110 ± 0.005	0.124 ± 0.009	0.076 ± 0.003	0.239 ± 0.008
Top-K Raw	0.011 ± 0.012	0.636 ± 0.020	0.465 ± 0.008	-0.002 ± 0.030	0.302 ± 0.005	0.196 ± 0.005	0.268 ± 0.007
Full Raw	-0.068 ± 0.002	0.638 ± 0.008	0.482 ± 0.001	-0.022 ± 0.013	$\mathbf{0.344 \pm 0.004}$	$\mathbf{0.421 \pm 0.017}$	0.299 ± 0.003
ReHarness	$\mathbf{0.234 \pm 0.014}$	0.500 ± 0.009	$\mathbf{0.515 \pm 0.009}$	$\mathbf{0.437 \pm 0.011}$	0.198 ± 0.017	0.016 ± 0.001	$\mathbf{0.317 \pm 0.004}$
LLM: Gemini 3.7 Flash							
Mem0	0.190 ± 0.007	0.066 ± 0.010	0.271 ± 0.010	-0.337 ± 0.014	0.357 ± 0.007	0.003 ± 0.002	0.092 ± 0.005
RBank	-0.014 ± 0.003	0.058 ± 0.010	0.017 ± 0.007	0.182 ± 0.005	0.000 ± 0.001	-0.003 ± 0.005	0.040 ± 0.002
DoT	-0.124 ± 0.002	0.247 ± 0.013	0.077 ± 0.005	-0.249 ± 0.011	0.014 ± 0.006	-0.011 ± 0.003	-0.008 ± 0.004
ACE	-0.047 ± 0.003	0.129 ± 0.003	0.089 ± 0.004	0.085 ± 0.014	0.000 ± 0.001	-0.038 ± 0.004	0.036 ± 0.003
GEPA-Online	0.039 ± 0.004	0.286 ± 0.005	$\mathbf{0.459 \pm 0.009}$	0.041 ± 0.004	-0.043 ± 0.005	0.012 ± 0.004	0.132 ± 0.007
Top-K Raw	$\mathbf{0.209 \pm 0.018}$	$\mathbf{0.701 \pm 0.012}$	0.365 ± 0.007	-0.356 ± 0.029	0.222 ± 0.003	-0.004 ± 0.009	0.190 ± 0.006
Full Raw	-0.036 ± 0.002	0.440 ± 0.009	0.426 ± 0.003	$\mathbf{0.343 \pm 0.002}$	0.160 ± 0.007	$\mathbf{0.070 \pm 0.011}$	$\mathbf{0.234 \pm 0.003}$
ReHarness	0.044 ± 0.010	0.610 ± 0.004	0.432 ± 0.007	-0.141 ± 0.024	$\mathbf{0.458 \pm 0.009}$	-0.005 ± 0.002	0.233 ± 0.003

Table 6: Normalized gain for each LLM on each MemoryBench task, with the highest value in bold.

Method	HB-Academic	HB-Creative	WB-Academic	WB-Creative	WB-Politics	Average
LLM: Claude Opus 4.8						
Mem0	0.073 ± 0.004	$\mathbf{0.182 \pm 0.009}$	0.000 ± 0.004	0.053 ± 0.003	-0.065 ± 0.009	0.049 ± 0.002
RBank	-0.055 ± 0.004	0.111 ± 0.004	0.026 ± 0.002	0.033 ± 0.003	0.000 ± 0.003	0.023 ± 0.002
DoT	0.077 ± 0.002	0.179 ± 0.004	-0.013 ± 0.002	0.047 ± 0.003	-0.026 ± 0.003	0.053 ± 0.003
ACE	0.115 ± 0.004	0.134 ± 0.001	0.064 ± 0.007	0.113 ± 0.005	0.000 ± 0.004	0.085 ± 0.004
GEPA-Online	0.166 ± 0.005	0.132 ± 0.002	0.090 ± 0.029	$\mathbf{0.173 \pm 0.003}$	$\mathbf{0.130 \pm 0.006}$	0.138 ± 0.013
Top-K Raw	0.118 ± 0.003	-0.008 ± 0.004	0.026 ± 0.002	0.040 ± 0.002	0.000 ± 0.002	0.035 ± 0.001
Full Raw	0.122 ± 0.003	0.167 ± 0.003	0.000 ± 0.003	0.080 ± 0.005	-0.052 ± 0.002	0.063 ± 0.002
ReHarness	$\mathbf{0.199 \pm 0.009}$	0.158 ± 0.002	$\mathbf{0.141 \pm 0.006}$	$\mathbf{0.173 \pm 0.006}$	$\mathbf{0.130 \pm 0.005}$	$\mathbf{0.160 \pm 0.003}$
LLM: GPT-5.5						
Mem0	0.094 ± 0.003	0.070 ± 0.008	-0.016 ± 0.005	0.000 ± 0.003	-0.066 ± 0.005	0.016 ± 0.003
RBank	-0.009 ± 0.005	0.101 ± 0.006	-0.032 ± 0.002	0.009 ± 0.003	-0.049 ± 0.003	0.004 ± 0.003
DoT	0.100 ± 0.003	0.092 ± 0.002	-0.016 ± 0.004	0.009 ± 0.004	-0.049 ± 0.003	0.027 ± 0.002
ACE	-0.008 ± 0.002	0.071 ± 0.004	-0.063 ± 0.005	0.009 ± 0.003	-0.098 ± 0.004	-0.018 ± 0.004
GEPA-Online	0.163 ± 0.007	0.139 ± 0.002	0.032 ± 0.007	0.000 ± 0.009	-0.016 ± 0.006	0.064 ± 0.006
Top-K Raw	0.169 ± 0.001	0.051 ± 0.007	-0.032 ± 0.003	0.035 ± 0.004	-0.016 ± 0.002	0.041 ± 0.001
Full Raw	0.157 ± 0.007	0.102 ± 0.003	-0.016 ± 0.003	-0.009 ± 0.006	-0.115 ± 0.004	0.024 ± 0.003
ReHarness	$\mathbf{0.215 \pm 0.010}$	$\mathbf{0.172 \pm 0.002}$	$\mathbf{0.048 \pm 0.005}$	$\mathbf{0.044 \pm 0.010}$	$\mathbf{0.000 \pm 0.007}$	$\mathbf{0.096 \pm 0.006}$
LLM: Gemini 3.7 Flash						
Mem0	$\mathbf{0.135 \pm 0.002}$	0.026 ± 0.010	$\mathbf{0.076 \pm 0.004}$	-0.045 ± 0.004	$\mathbf{0.071 \pm 0.001}$	0.053 ± 0.002
RBank	0.031 ± 0.003	$\mathbf{0.193 \pm 0.004}$	0.000 ± 0.002	-0.025 ± 0.004	0.024 ± 0.004	0.045 ± 0.003
DoT	0.045 ± 0.002	0.094 ± 0.006	-0.076 ± 0.001	-0.025 ± 0.003	0.012 ± 0.002	0.010 ± 0.002
ACE	0.098 ± 0.004	0.069 ± 0.002	-0.013 ± 0.003	0.000 ± 0.004	-0.012 ± 0.004	0.028 ± 0.005
GEPA-Online	-0.062 ± 0.010	0.119 ± 0.002	-0.772 ± 0.010	$\mathbf{0.032 \pm 0.005}$	0.012 ± 0.007	-0.134 ± 0.010
Top-K Raw	0.113 ± 0.002	0.131 ± 0.006	-0.013 ± 0.002	-0.032 ± 0.003	-0.012 ± 0.002	0.037 ± 0.001
Full Raw	-0.032 ± 0.006	0.153 ± 0.003	-0.063 ± 0.002	-0.064 ± 0.005	-0.060 ± 0.004	-0.013 ± 0.003
ReHarness	0.040 ± 0.009	0.173 ± 0.002	0.013 ± 0.004	$\mathbf{0.032 \pm 0.002}$	0.060 ± 0.004	$\mathbf{0.064 \pm 0.003}$

Table 7: Normalized gain for each LLM on each MemoryArena task, with the highest value in bold.

Method	Travel	Shopping	Search	Math	Physics	Average
LLM: Claude Opus 4.8						
Mem0	$\mathbf{0.451 \pm 0.022}$	0.029 ± 0.028	-0.057 ± 0.003	0.016 ± 0.002	0.024 ± 0.014	0.093 ± 0.005
RBank	0.037 ± 0.003	0.043 ± 0.002	0.019 ± 0.002	-0.020 ± 0.003	-0.010 ± 0.004	0.014 ± 0.001
DoT	0.085 ± 0.004	0.121 ± 0.010	0.124 ± 0.002	0.013 ± 0.004	0.300 ± 0.008	0.129 ± 0.008
ACE	-0.002 ± 0.001	0.079 ± 0.001	0.048 ± 0.004	0.012 ± 0.002	0.089 ± 0.007	0.045 ± 0.003
GEPA-Online	0.009 ± 0.004	0.219 ± 0.014	0.105 ± 0.007	0.077 ± 0.002	0.218 ± 0.007	0.126 ± 0.005
Top-K Raw	0.292 ± 0.022	0.138 ± 0.007	-0.514 ± 0.011	0.148 ± 0.001	$\mathbf{0.663 \pm 0.017}$	0.145 ± 0.007
Full Raw	0.274 ± 0.008	0.179 ± 0.012	-0.819 ± 0.027	$\mathbf{0.192 \pm 0.002}$	0.582 ± 0.005	0.082 ± 0.006
ReHarness	0.443 ± 0.022	$\mathbf{0.221 \pm 0.008}$	$\mathbf{0.210 \pm 0.007}$	0.138 ± 0.003	0.511 ± 0.007	$\mathbf{0.305 \pm 0.003}$
LLM: GPT-5.5						
Mem0	0.771 ± 0.013	0.466 ± 0.024	0.000 ± 0.003	0.103 ± 0.002	0.271 ± 0.008	0.322 ± 0.010
RBank	0.000 ± 0.003	0.011 ± 0.003	0.014 ± 0.002	0.010 ± 0.003	0.048 ± 0.003	0.017 ± 0.001
DoT	0.176 ± 0.007	0.383 ± 0.006	0.060 ± 0.003	0.119 ± 0.006	0.155 ± 0.009	0.179 ± 0.005
ACE	-0.012 ± 0.001	0.059 ± 0.002	0.009 ± 0.004	0.015 ± 0.002	-0.051 ± 0.007	0.004 ± 0.003
GEPA-Online	-0.149 ± 0.005	0.327 ± 0.008	$\mathbf{0.321 \pm 0.003}$	-0.011 ± 0.006	0.155 ± 0.007	0.129 ± 0.002
Top-K Raw	0.781 ± 0.002	0.425 ± 0.013	0.023 ± 0.020	0.150 ± 0.001	0.319 ± 0.015	0.340 ± 0.009
Full Raw	0.787 ± 0.023	$\mathbf{0.471 \pm 0.010}$	0.014 ± 0.022	$\mathbf{0.168 \pm 0.003}$	0.350 ± 0.011	$\mathbf{0.358 \pm 0.009}$
ReHarness	$\mathbf{0.816 \pm 0.024}$	0.407 ± 0.008	0.101 ± 0.007	0.115 ± 0.004	$\mathbf{0.351 \pm 0.010}$	$\mathbf{0.358 \pm 0.006}$
LLM: Gemini 3.7 Flash						
Mem0	0.904 ± 0.010	0.005 ± 0.031	-0.030 ± 0.003	0.089 ± 0.003	0.209 ± 0.002	0.235 ± 0.008
RBank	-0.005 ± 0.002	0.037 ± 0.002	0.045 ± 0.001	-0.031 ± 0.003	0.069 ± 0.004	0.023 ± 0.001
DoT	0.009 ± 0.004	0.072 ± 0.017	0.082 ± 0.002	0.013 ± 0.005	0.035 ± 0.007	0.042 ± 0.004
ACE	-0.009 ± 0.001	0.058 ± 0.001	0.090 ± 0.002	0.048 ± 0.002	0.180 ± 0.008	0.073 ± 0.004
GEPA-Online	0.000 ± 0.007	0.070 ± 0.012	0.201 ± 0.004	-0.008 ± 0.003	0.028 ± 0.008	0.058 ± 0.004
Top-K Raw	0.940 ± 0.014	0.221 ± 0.013	-0.269 ± 0.014	$\mathbf{0.156 \pm 0.001}$	0.407 ± 0.013	0.291 ± 0.007
Full Raw	$\mathbf{0.945 \pm 0.017}$	0.182 ± 0.008	-0.537 ± 0.016	0.142 ± 0.002	0.452 ± 0.010	0.237 ± 0.010
ReHarness	0.930 ± 0.021	$\mathbf{0.226 \pm 0.009}$	$\mathbf{0.291 \pm 0.005}$	0.086 ± 0.002	$\mathbf{0.508 \pm 0.004}$	$\mathbf{0.408 \pm 0.003}$

You are DecisionAgent. The input contains ranked reflections from previous instances, followed by the current task and observations.

Choose the next structured action for the current task. Return only one JSON object matching the supplied schema. Do not include private reasoning in the JSON.

The retrieved memory is supplied separately as:

[RANKED REFLECTIONS]
[RANK 1 OF K REFLECTION]
{reflection}
…

{current task and observations}

You are ReflectionAgent. The input contains the current task, trajectory, and outcome, and may include semantically similar reflection examples.

Write a concise reflection that will help maximize reward on similar future tasks. Use separate bullet lists for (1) consequential operational evidence, including what worked, what failed, and any recovery, and (2) strongly supported interpretations and reusable reward-maximizing strategies. Clearly distinguish evidence from interpretation.

Examples show writing style and perspectives already covered. Prefer a supported perspective that adds diversity, but do not fabricate novelty.

Return only JSON exactly as {"reflection":"..."}.

Input layout:

[REVISED REFLECTION DEMONSTRATIONS]
[DEMONSTRATION 1]
{revised reflection}
…

[CURRENT TASK, TRAJECTORY, AND OUTCOME]
{completed trajectory}

You are RevisionAgent. The input contains one reflection and its guidance, its source trajectory, and a later trajectory in which it was retrieved.

Revise the reflection so it will help maximize reward on similar future tasks. Use separate bullet lists for (1) consequential operational evidence, including what worked, what failed, and any recovery, and (2) strongly supported interpretations and reusable reward-maximizing strategies. Clearly distinguish evidence from interpretation.

Write concise use-case guidance as bullet points describing the matching conditions under which specific evidence or strategies in the reflection apply. If the guidance is empty, initialize it; otherwise refine it using the later trajectory.

Use the source trajectory as the sole source of content. Use the later trajectory only to refine matching conditions and identify omissions supported by the source trajectory. Do not import content from the later trajectory or attribute its behavior to the target reflection.

Return only JSON exactly as {"revised_reflection":"..."}.

Input layout:

[TARGET MEMORY ID]
{memory_id}

[CURRENT REFLECTION]
{reflection}

[TARGET MEMORY'S ORIGINAL TRAJECTORY]
{source trajectory}

[LATEST OBSERVED USE]
{current trajectory}

Figure 5: Prompts used by the decision agent π_D (top), memory agent π_M (middle), and revision agent π_R (bottom) in ReHarness.

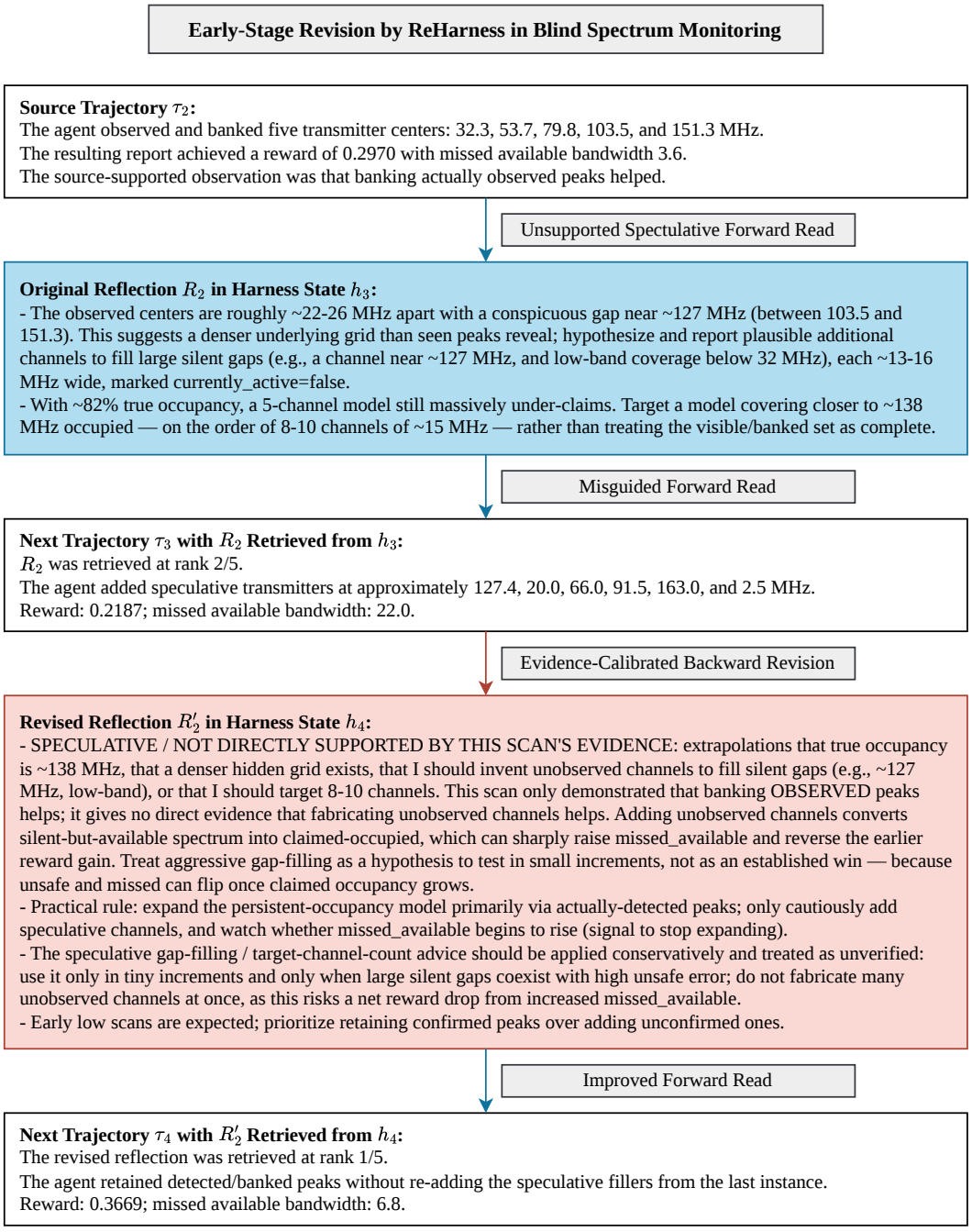

Figure 6: Full qualitative revision example for blind spectrum monitoring from Figure 3 (a).

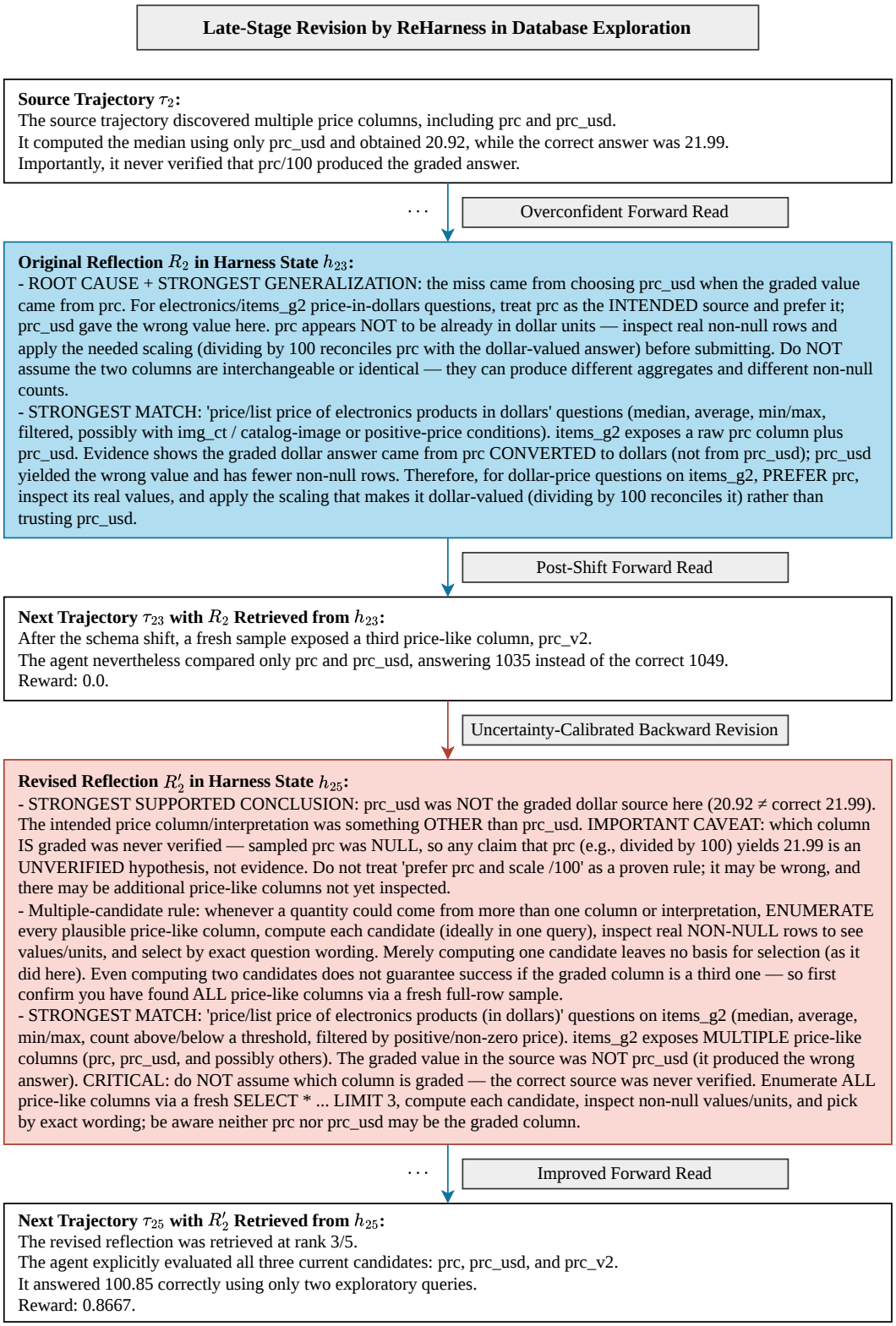

Figure 7: Full qualitative revision example for database exploration from Figure 3 (b).