ORGMEM: BENCHMARKING SELF-EVOLVING AGENTS FOR ORGANIZATION-SPECIFIC EVOLVING AUDIT ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents increasingly rely on external memory to reuse experience across long-horizon interactions, yet existing evaluations provide limited insight into whether historical experience remains useful when the underlying decision environment evolves. This challenge is particularly important in organizational auditing, where procurement policies, market conditions, vendor behavior, and fraud-risk patterns may change over time, causing superficially similar cases to require different judgments. We introduce AuditEvo, a synthetic benchmark for evaluating self-evolving agents under organization-specific and implicitly evolving procurement environments. AuditEvo contains 40,000 evaluated cases organized into 100 chronological organizational chains with evolving policy, market, vendor, and risk conditions. At each step, an agent observes the current case and chronologically available history, predicts an audit outcome of approval, rejection, or investigation, and immediately receives verified feedback that may be reused in subsequent decisions. A deterministic audit oracle provides reproducible labels while keeping latent environment states and oracle rationales hidden from evaluated agents. Experiments across diverse memory and reflection frameworks and multiple LLM backbones show that current systems struggle particularly with risk-regime cases and that retrieving historically related cases alone does not ensure reliable decisions when their applicability changes. Motivated by this limitation, we introduce Environment-Aware JitRL, which maintains an evolving playbook that uses feedback to adapt state similarity and thereby condition historical experience retrieval for action-value estimation. Across evaluated backbones, the method improves average audit accuracy over JitRL by approximately 2–7 percentage points. AuditEvo provides a controlled testbed for studying memory reuse, adaptation, and negative transfer under persistent multi-environment evolution.

1 INTRODUCTION

Large language model (LLM) agents increasingly rely on external memory to support decisions across extended interactions. Recent memory frameworks (Packer et al., 2023; Xu et al., 2026b; Chhikara et al., 2025) preserve information beyond the immediate context window and organize accumulated experience for subsequent use. In real-world business workflows, however, the conditions governing the usefulness of past experience evolve over time including policies, markets, and firm behaviors etc. A historical case may therefore closely resemble a current transaction while supporting an inappropriate decision under the current regime. Reliable memory therefore requires agents to extract transferable patterns from past experience and assess whether those patterns remain applicable under current conditions. Established agent benchmarks (Liu et al., 2024; Merrill et al., 2026; Boisvert et al., 2024) primarily evaluate task completion under predefined task and environment configurations. Recent benchmarks extend this evaluation by examining experience reuse across task streams (Wei et al., 2025) and adaptation to persistent environmental changes in terminal, software, and personalization domains (Xu et al., 2026a). A complementary challenge arises in finance, where policy, market, and operational risk environments evolve within the same organizational workflow. Their evolution affects the applicability of historical experience, while useful precedents may involve different vendors or transactions. For example, an agent may need to draw on resolved

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.

concealed-ownership cases involving earlier suppliers to recognize a similar pattern among new vendors, while determining whether the historical evidence and reasoning remain applicable under current conditions. Evaluating this capability requires temporally connected cases that combine evolving decision conditions with feedback from historical outcomes. This motivates a benchmark that jointly evaluates inference of underlying regimes, transfer of similar patterns across cases, and selective reuse of experience throughout an organization's evolving decision history.

To address this gap, we introduce **AuditEvo**, a benchmark to evaluate agents in implicitly evolving procurement-audit environments. AuditEvo comprises 40,000 evaluated cases organized into 100 chronological organizational chains, each spanning at least three fiscal years. Within each chain, policy, market, and organizational risk environments evolve through policy document revisions, changes in prices and vendor behaviors. Agents must infer the evolving conditions through comparing documents and analyzing accumulated transaction evidence. Each organization has its own procurement rules and operational conventions. AuditEvo couples this multi-environment evolution with experience reuse from resolved cases and their ground-truth outcomes. At each step, an agent receives the current case and applicable policy, accesses earlier resolved cases, and predicts approval, rejection, or investigation. The current outcome is withheld until prediction and can subsequently enter memory as feedback for later decisions. Historical outcomes also provide evidence about organization-specific audit conventions, allowing the agent to infer how comparable circumstances were judged before the current case arrives. A deterministic audit oracle provides reproducible labels while keeping latent benchmark state inaccessible to the agent. Scenarios such as split purchasing, emergency abuse, and bid rotation distribute decisive evidence across earlier cases, requiring agents to connect multiple observations. Table 1 displays comparison between AuditEvo and related agent benchmarks.

We evaluate AuditEvo on a wide range of agent frameworks alongside a no-memory baseline across multiple open- and closed-source backbone models. The results indicate that most current agent frameworks struggle to uncover risk pattern and give reliable audit decisions as environments evolve. Among them, JitRL (Li et al., 2026c) achieves better performance in most models. However, its fixed state representation design limits its capacity to adapt to new environments. To address this challenge, we propose Environment-Aware JitRL, which extends JitRL by introducing an evolving playbook to adapt historical experience retrieval and reasoning under changing environmental conditions. Empirically, our method improve audit decision-making by 2% – 7%. Overall, our contributions are threefold:

- **A benchmark for persistent multi-environment evolution with online feedback.** We introduce AuditEvo, which combines organization-specific procurement workflows with evolving policy, market, vendor, and fraud-risk conditions across chronological case chains, while providing immediate verified outcomes for subsequent experience reuse.
- **A systematic evaluation of agent memory under changing experience applicability.** We benchmark representative retrieval, memory-management, reflection, and runtime-learning frameworks across multiple LLM backbones and evaluate both audit decisions and overlap with annotated historical references, revealing substantial limitations in selective experience reuse under evolving conditions.
- **Environment-aware experience reuse.** We introduce Environment-Aware JitRL, which uses an evolving playbook to adapt state similarity from immediate feedback and retrieve historical experiences that are more compatible with the current decision context before performing JitRL-style action-value estimation.

2 Related Work

2.1 Agent Memory

Retrieval-augmented generation (RAG) (Lewis et al., 2020; Yu et al., 2023; Borgeaud et al., 2022) provides a basic form of external memory by retrieving semantically relevant text from a stored corpus, but typically treats historical information as static chunks without explicitly deciding what should be retained, revised, or forgotten. Agent-memory research extends this idea toward persistent and adaptive memory management. MemoryBank (Zhong et al., 2024) organizes interaction history

Benchmark	Domain	What Evolves?	PE	IC	OF	MR
AgentBench (Liu et al., 2024)	Multi-env.	Static tasks	×	×	×	×
Terminal-Bench(Merrill et al., 2026)	Terminal	Static tasks	×	×	×	×
WorkArena++(Boisvert et al., 2024)	Workflow	Static tasks	×	×	×	×
Finance Agent (Bigeard et al., 2025)	Finance	Static Tasks	×	×	×	×
Evo-Memory (Wei et al., 2025)	Multi-task	User memory	×	△	✓	✓
SWE-bench-Live (Li et al., 2026a)	Software	Task refresh	×	×	×	×
GAIA2 (Froger et al., 2026)	Tool / Web	Async events	△	✓	×	×
LongMemEval(Wu et al., 2024)	Conversation	Personal Information	△	△	×	✓
HorizonBench(Li et al., 2026b)	Personalization	Preference changes	△	✓	×	✓
EvoArena(Xu et al., 2026a)	Terminal, Software, Persona	Dynamic Env	✓	✓	×	△
AuditEvo	**Financial Audit**	**Multi-Dynamic Env**	✓	✓	✓	✓

Table 1: **Comparison of AuditEvo with related agent benchmarks.** PE denotes persistent environment evolution, where the same environment or user state evolves across versions; IC denotes implicit change, where the agent must infer or handle changes from context or interaction; OF denotes online outcome feedback: after completing a task, the agent receives an externally verified outcome that can inform memory updates and subsequent decisions. MR denotes multi-session reasoning. ✓ denotes supported, × denotes not supported, and △ denotes partial support.

through hierarchical summaries and a forgetting mechanism, while MemGPT (Packer et al., 2023) separates limited working context from larger external memory. More recent approaches dynamically restructure stored experience: A-MEM (Xu et al., 2026b) represents memories as interconnected notes enriched with LLM-generated attributes and evolving links, whereas Mem0 (Chhikara et al., 2025) maintains salient facts through explicit ADD, UPDATE, DELETE, and NOOP operations. Runtime-learning methods further incorporate feedback into memory utility. MemRL (Zhang et al., 2026b) uses two-stage retrieval that first filters memories by semantic relevance and then ranks them using utility values updated from environmental feedback, enabling non-parametric improvement without modifying model weights. JitRL (Li et al., 2026c) similarly maintains a dynamic non-parametric experience memory, but retrieves relevant trajectories to estimate action advantages at test time and directly adjusts the LLM's action distribution without gradient updates. Collectively, these methods illustrate the evolution of agent memory from static semantic retrieval toward adaptive systems that modify memory content, structure, or utility based on accumulated experience.

2.2 Dynamic and Evolving Financial Agent Benchmarks

Dynamic evaluation is particularly important in finance because real-world decision environments are inherently non-stationary, whereas many existing benchmarks remain largely static. FinanceBench (Islam et al., 2023), FinBen (Xie et al., 2024), and FinDABench (Liu et al., 2025) evaluate financial question answering, analysis, forecasting, and decision-making over fixed task sets. More recent benchmarks introduce agentic and temporal interaction: Finance Agent Benchmark (Bigeard et al., 2025) and FinAgentBench (Choi et al., 2025) assess tool use and multi-step retrieval, InvestorBench (Li et al., 2025) evaluates interactive financial decision-making, and StockBench (Chen et al., 2025) and RISCTrade (Huang et al., 2026) study sequential trading under changing market conditions. However, these settings primarily capture market evolution or repeated interaction rather than persistent changes in the governing decision environment itself. In practice, agents must also adapt to evolving regulations, organizational policies, fraud patterns, and other contextual conditions that can render prior experience obsolete or only partially applicable. This motivates benchmarks that organize cases into temporally ordered regimes and explicitly evaluate whether agents can detect environmental change, preserve invariant knowledge, discount incompatible memories, and adapt reasoning over time.

3 AuditEvo: An Evolving Procurement Audit Benchmark

We introduce **AuditEvo**, a synthetic benchmark for evaluating long-horizon AI agents and memory systems under continuously evolving procurement audit environments. Unlike conventional audit benchmarks that treat cases as independent observations under a fixed set of rules and risk conditions,

AuditEvo organizes procurement activities into temporally ordered environment chains. Within each chain, an auditing agent sequentially reviews procurement cases while the governing purchasing policy, vendor behavior, vendor identity relationships, market conditions, and fraud-risk patterns evolve over time. Consequently, the correct assessment of a current case may depend not only on its contemporaneous transaction information and applicable purchasing policy, but also on relevant evidence accumulated from earlier procurement cases. AuditEvo is generated through a simulator with deterministic audit logics, enabling reproducible ground-truth audit opinions while preserving long-horizon dependencies, latent environmental changes, and strict separation between agent-visible evidence and hidden oracle information.

Formally, each chain j can be represented as a sequence

$$\mathcal{C}^{(j)} = [(E_1, c_1), (E_2, c_2), \ldots, (E_T, c_T)] \tag{1}$$

where c_t denotes the procurement case generated at temporal step t and E_t denotes the contemporaneous environment state. The environment contains the active procurement policy, vendor states, market-price conditions, organizational structure, and the resolved historical cases available before the current decision. Importantly, E_t is not explicitly exposed to the evaluated agent. Some changes like rules are communicated through observable artifacts such as a newly issued policy document, whereas other changes (e.g. fraud pattern, vendor state, market conditions) are latent and must be inferred from prior procurement observations. Internally, the generator records these transitions as environment patches with the pre-change state, post-change state, affected entities, event time, change type etc.

We accumulate real-world private/public procurement data from local governments, universities, and firms and generate our synthetic data based on them. Each chain begins by sampling an organization, its departments, a vendor ecosystem, an initial purchasing policy, product and service categories, market-price reference levels, and department-specific payment-timing profiles. The current implementation supports heterogeneous organizational contexts, including universities, hospitals, municipalities, manufacturing companies, and nonprofit organizations, with procurement categories aligned to the functions of individual departments. Each chain initially contains approximately 10-18 vendors and is required to span at least three fiscal years, ensuring that long-horizon evaluation includes multiple policy periods rather than a single static procurement regime.

3.1 Case Representation

A procurement case is presented as an LLM-naturalized textual paragraph describing the purchasing department, vendor, product or service, quantity, price, requested and invoiced amounts, procurement method, and other transaction attributes, together with structured metadata such as the case identifier and event date. The current case and its preceding historical cases jointly provide evidence about an underlying latent scenario that the agent must infer through its memory mechanism. This temporal structure is essential because many procurement risks cannot be identified from an isolated transaction. For example, delayed delivery is reflected in the elapsed time between order and delivery, premature invoicing in the relative ordering of invoice and delivery events, and abnormal payment behavior in deviations from recent department-specific payment patterns. The generator therefore produces temporally linked operational trajectories rather than explicitly providing precomputed risk indicators. For each audit task, the agent must interpret the current case using its general knowledge together with relevant prior experience stored in memory. For instance, a split-purchase case is preceded by related approved purchases; a procurement-card misuse episode is built from preceding card transactions; payment anomalies are evaluated relative to recent normal same-department payment cases. To prevent look-ahead bias, each case is associated with a retrieval cutoff that restricts accessible memory to events occurring strictly before the current case. Thus, for a case c_t, the admissible history is $H_t = (e_i \mid i < t)$, rather than the complete chain. The public record identifies the corresponding chain and event index, allowing retrieval to be constrained causally and preventing future cases from leaking into earlier decisions.

3.2 Policy Environment Evolution

A defining characteristic of AuditEvo is that the governing policy is not fixed throughout a chain. Each chain begins with an independently sampled purchasing policy containing an an-

nual vendor-department threshold, department-specific threshold overrides, organization-wide bid thresholds, minimum quotation requirements, a high-value purchase-order threshold, fiscal-year definition, and rules specifying which purchasing instruments contribute to cumulative vendor spending. These policy variables are rendered into natural-language purchasing-procedure documents that resemble artifacts an agent might receive in an operational procurement workflow. Policy evolution occurs at fiscal-year boundaries. When a new fiscal year is reached, the simulator creates a new policy version and modifies one or more policy operators. In the default configuration, each revision changes one or two of four policy dimensions: $\mathcal{O}_P = \{$*department threshold*, *bid thresholds*, *high-value PO threshold*, *quotation requirement*$\}$ Crucially, the agent is not provided with an explicit policy diff such as "the threshold changed from $50,000 to $40,000". Instead, a complete replacement purchasing-procedure document is published and attached to cases governed by the new version. This design creates a form of artifact-visible but change-implicit evolution: the existence of a new policy document is observable, whereas identifying exactly what changed requires comparison with earlier policy knowledge. The policy governing case c_t is selected according to the case's request date, rather than simply using the most recently generated document. If $\mathcal{P} = \{P_1, \ldots, P_K\}$ denotes the policy history and d_t^{req} is the request date of c_t, the applicable policy can be written as

$$P(c_t) = \underset{P_k:\, d(P_k) \le d_t^{req}}{\arg\max} \; d(P_k) \tag{2}$$

This date-aware design prevents a common benchmark shortcut in which the latest rule is blindly applied to every historical observation. Policy violations are generated across several dimensions, including use of an incorrect bid process, insufficient competitive quotations, an impermissible purchasing instrument for a high-value purchase, and violation of the applicable cumulative vendor–department threshold. Because these rules themselves evolve, structurally similar cases may legitimately receive different audit outcomes at different points in the chain.

3.3 Risk Environment Evolution

In a corporate governance context, the risk environment refers to internal conditions, operational vulnerabilities, and behavioral patterns that may facilitate or increase the likelihood of fraud. We model changes in this risk environment in addition to formal policy evolution. Rather than assigning fraud as an independent binary attribute to individual transactions, the benchmark represents risk through observable consequences, temporally linked behavior, and latent state changes. We simulate a range of procurement fraud and risk patterns, including premature invoicing, abnormal payment timing, split purchasing, procurement-card misuse, emergency-purchase abuse, sole-source vendor concentration, and bid-rotation collusion, with different patterns emerging over time. Importantly, agents are not explicitly informed that a vendor has become risky or that a new fraud regime has begun. Instead, they must infer such changes from evidence distributed across multiple cases using their memory mechanisms. Specifically, individually benign or non-actionable anchor cases precede a later target case whose interpretation depends on the accumulated sequence. For example, split purchasing is revealed only after a series of related purchases; procurement-card misuse and emergency-purchase abuse similarly emerge through repeated transactions; sole-source concentration requires multiple awards to the same vendor; and bid rotation is identified from recurring award sequences across different vendors. For instance, the benchmark may evaluate an $A-B-C-A-B-C$ award pattern within a temporal lookback window rather than inferring collusion from a single transaction. Under balanced generation, each chain contains both clean anchor cases and targeted policy or risk cases. To reduce ambiguity while preserving long-horizon reasoning, each risk-target case is associated with a single primary fraud or risk family.

3.4 Macro Environment Evolution

Real-world business operations are strongly influenced by external conditions beyond the organization, particularly changes in suppliers, markets, and inter-firm relationships. To capture this non-stationarity, AuditEvo models several forms of external environment evolution.

Latent vendor-behavior change Vendor behavior evolves independently of formal procurement policy. At an environment transition, a vendor's delivery-delay probability or price multiplier may increase or decrease. These latent parameters are hidden from the agent and become observable only

through subsequent delivery outcomes and transaction prices. The agent must therefore infer vendor deterioration or improvement from accumulated historical evidence rather than relying on an explicit vendor-risk label.

Latent market change The simulator maintains category-specific market-price conditions that may shift upward or downward over time. These market states are not directly revealed to the agent and can only be inferred from prices observed across multiple vendors and transactions. This distinction is important for price-risk assessment, because an unusually high transaction price should not necessarily be interpreted as vendor-specific inflation when similar increases occur across the broader market. The agent must therefore compare the current case with temporally relevant peer transactions rather than rely on a static historical price benchmark.

Latent vendor-identity evolution AuditEvo also models changes in supplier relationships through hidden affiliation groups. During a concealed-ownership transition, previously independent vendors may become affiliated and subsequently share observable attributes such as addresses or contact information. The underlying beneficial-ownership relationship remains hidden, requiring the agent to infer emerging entity connections from evidence accumulated across multiple cases rather than from an explicitly provided ownership graph.

3.5 Deterministic Audit Oracle and Ground-Truth Construction

A central design objective of AuditEvo is to separate data generation from model-based annotation. Rather than asking an LLM to judge whether a generated procurement case is compliant or suspicious, the benchmark uses a deterministic audit oracle verified by human expert. For every generated case, the oracle retrieves the policy effective on the request date, reconstructs the relevant approved historical purchase set, computes applicable cumulative spending, determines the required bid procedure and minimum quotation count, verifies the purchasing instrument etc. Let $V_t = \{v_{t,1}, \ldots, v_{t,m}\}$ denote the policy violations identified for case c_t, and $R_t = \{r_{t,1}, \ldots, r_{t,n}\}$ denote the fraud or vendor-risk signals. The retrospective audit label is deterministically assigned as

$$y_t = \begin{cases} \texttt{investigation}, & |R_t| > 0, \\ \texttt{reject}, & |R_t| = 0 \ \wedge \ |V_t| > 0, \\ \texttt{approve}, & |R_t| = 0 \ \wedge \ |V_t| = 0. \end{cases} \tag{3}$$

This precedence rule intentionally distinguishes suspicious cases requiring investigation from deterministic policy noncompliance. The oracle additionally records the active policy version, cumulative amount before and after the case, applicable and excluded historical cases, supporting evidence identifiers, specific violations and risk signals, and an auditable rationale. For evaluation, the generator constructs an oracle-reference set for each task by combining applicable spending history, oracle detector evidence, risk-support cases, and calibration precedents, restricted to chronologically eligible prior cases. This set records the historical references selected by the benchmark's construction procedure. It is neither a minimal sufficient supporting set nor an exhaustive collection of all valid evidence. Consequently, alternative historical cases may support a correct decision without belonging to the reference set. These reference annotations remain hidden from the evaluated agents. AuditEvo incorporates a validation layer after case generation and oracle evaluation. Quality assurance focuses on internal logical consistency, causal validity, scenario realization, and separation between observable and oracle-only information. The appendix details the quality control check.

We evaluate agents under a chronological online protocol in which each procurement case is encountered and evaluated only once. Before predicting the current case x_t, the agent may access only previously resolved cases and their feedback through memory $\mathcal{M}_t$, while the current ground-truth outcome $y_t \in \{\textit{approve}, \textit{investigation}, \textit{reject}\}$ remains hidden until the prediction $\hat{y}_t$ is finalized. The case and its verified outcome are then immediately incorporated into memory for subsequent decisions. This protocol enables agents to learn organization-specific decision patterns from historical experience while preventing look-ahead leakage; latent environment states and organization-specific parameters remain hidden and must be inferred from observable historical evidence.

4 EXPERIMENTS

4.1 DATASET

We synthesize 40,000 evaluated procurement cases organized into 100 chronological chains, each comprising 400 cases. The evaluated cases consist of 35% normal cases (approve; 14,000 cases), 30% policy-violation cases (reject; 12,000 cases), and 35% risky cases (investigation; 14,000 cases). We also add 31 calibration cases per chain. They guarantee that the first risk cases are identifiable and subsequent scored cases create additional references naturally. The detailed summary statistics of dataset is provided in appendix A.4.

4.2 EXPERIMENTAL SETUP

Agents and Models We evaluate No Memory setting along with a series of agent memory frameworks including Standard RAG (Lewis et al., 2020), MEM0 (Chhikara et al., 2025), MemoryBank (Zhong et al., 2024), A-MEM (Xu et al., 2026b), ACE (Zhang et al., 2026a), Reflexion (Shinn et al., 2023), Meta-Reflexion (Wu et al., 2025), MemRL (Zhang et al., 2026b), JitRL (Li et al., 2026c) on our benchmark AuditEvo. We also pair each framework with multiple open/private backbone models different in size, including Llama3.2-3b (AI@Meta, 2024), Llama3-8b (AI@Meta, 2024), Qwen2.5-3b (Team, 2025), Qwen3-8b (Team, 2025), GPT4o-mini (Achiam et al., 2023), GPT5(Singh et al., 2025).

Implementation Details For all methods, we maintain consistency by employing identical system prompts as detailed in Appendix. The deployment of open-source models is accomplished through local instantiation using vllm (Kwon et al., 2023). For GPT models, we employ the official structured output API. In general memory retrieval process,we primarily use k=10 for top-k memory selection to ensure comparisons between agent memory frameworks and maintain computational efficiency, while adjusting this parameter for specific components if they are important to optimize performance. For text embedding, we implement the all-minilm-l6-v2 model across all experiments.

Metrics We evaluate memory frameworks using audit decision performance and overlap with annotated historical references. We conduct a single chronological evaluation in which each scored case is predicted exactly once, and its ground-truth outcome becomes available only after the prediction is finalized. Audit-opinion accuracy is defined as $\mathrm{Acc} = \frac{1}{N}\sum_{t=1}^{N}\mathbb{I}(\hat{y}_t = y_t)$, where N is the number of evaluated cases. We additionally report macro-F1 by pooling predictions across cases and accuracy separately for normal, policy-only, and risk-only cases. To characterize the historical cases recovered by each memory strategy, we report oracle-reference case-overlap micro-F1, abbreviated as case-overlap F1. For each task t, let R_t denote the deduplicated source-case identifiers returned by the memory-strategy adapter and G_t the oracle-reference set defined in Section 3.5. The evaluated identifiers come from the adapter's retrieval output, rather than the model's final answer or citations. We aggregate $\mathrm{TP} = \sum_t |R_t \cap G_t|$, $\mathrm{FP} = \sum_t |R_t \setminus G_t|$, and $\mathrm{FN} = \sum_t |G_t \setminus R_t|$, and compute micro-F1 as $2\mathrm{TP}/(2\mathrm{TP} + \mathrm{FP} + \mathrm{FN})$. Here, false positives denote cases outside the annotated reference set and are not necessarily irrelevant evidence. For summarized or linked memories, source identifiers are reconstructed from available provenance according to the adapter procedures described in the appendix.

4.3 RESULTS

Table 2 shows that current agents struggle to maintain reliable audit decisions as policies and risk regimes evolve. Memory and reflection mechanisms generally improve performance on evolving cases, but the gains are inconsistent across models and regimes. Reinforcement-based approaches, particularly JitRL, tend to produce better performance, whereas retrieval- and reflection-based methods are often highly dependent on the underlying model. Case-overlap F1 remains modest, generally between the mid-20s and low-30s, and does not consistently align with audit accuracy or macro-F1. Although this metric captures only overlap with annotated reference cases, the weak association suggests that identifying historically related cases alone is insufficient, agents must also determine whether those cases remain applicable under the current conditions. Finally, more elaborate memory organizations often introduce substantial ingestion or retrieval costs without

Table 2: Experimental results on the AuditEvo dataset across three categories: Normal, Policy-Violation, and Risk-Regime. Results are reported in Acc (%) as well as average performance in Acc and F1. Case-overlap F1 denotes oracle-reference case-overlap micro-F1 computed from adapter-reported source-case identifiers; it measures overlap with the annotated reference sets rather than the completeness or sufficiency of retrieved evidence.

Model	Size	Method	Normal	Policy-Violation	Risk-Regime	Average			Retrieval Latency	Memory Ingest Latency
			Acc	Acc	Acc	Acc	Macro-F1	Case-Overlap F1		
GPT	4o-mini	No Memory	76.69	18.00	15.31	37.60	32.44	-	-	-
		RAG	66.96	62.68	5.90	44.31	38.89	25.50	0.027s	0.039s
		MEM0	69.38	51.55	5.97	41.84	36.57	25.62	0.147s	23.83s
		MemoryBank	71.19	55.52	4.88	43.28	37.37	28.80	0.125s	9.23s
		A-MEM	72.86	50.51	5.71	42.65	36.95	22.38	1.75s	7.69s
		Reflexion	59.32	56.90	10.63	41.55	38.26	25.22	0.07s	0.12s
		Meta-Reflexion	65.38	51.28	12.64	42.69	39.45	25.08	4.82s	0.04s
		ACE	55.59	58.73	6.27	39.27	35.07	25.16	0.03s	0.06s
		MemRL	46.50	36.33	34.00	39.08	39.00	28.16	0.07s	0.03s
		JitRL	**64.99**	**40.28**	**40.53**	**49.02**	**48.62**	**28.04**	**0.00s**	**0.00s**
		Ours	**69.98**	**39.55**	**48.42**	**53.31**	**55.10**	**30.14**	**0.09s**	**0.14s**
	5	No Memory	72.45	20.34	18.96	38.10	33.09	-	-	-
		RAG	68.24	69.54	9.88	48.20	43.94	25.50	0.03s	0.05s
		MEM0	69.04	65.49	10.54	47.50	42.37	27.43	0.08s	3.90s
		MemoryBank	69.24	64.00	8.99	46.58	41.09	29.04	0.07s	9.14s
		A-MEM	43.25	72.38	9.72	40.25	38.49	18.00	2.32s	49.67
		Reflexion	57.36	49.74	36.62	47.82	47.95	25.33	0.08s	0.13s
		Meta-Reflexion	65.38	65.70	17.03	48.55	45.90	25.08	16.48s	0.15s
		ACE	**60.00**	**58.92**	**38.55**	**52.17**	**52.19**	**25.43**	**0.06s**	**0.09s**
		MemRL	61.27	36.40	47.37	48.94	49.46	27.48	0.04s	0.02s
		JitRL	66.52	43.94	38.87	50.07	50.91	28.33	0.00s	0.00s
		Ours	**65.39**	**60.50**	**45.45**	**56.94**	**55.97**	**28.92**	**0.10s**	**0.12s**
Qwen	2.5-3B	No Memory	88.58	6.11	4.37	34.36	23.40	-	-	-
		RAG	61.87	54.71	17.68	44.25	42.47	25.10	0.00s	0.01s
		MEM0	82.16	0.50	35.47	41.32	31.32	27.41	0.01s	1.7s
		MemoryBank	81.20	1.34	33.65	40.60	30.03	28.43	0.00s	1.5s
		A-MEM	82.24	2.40	37.13	42.50	33.44	21.75	1.89s	3.16s
		Reflexion	81.98	29.88	9.11	40.85	34.86	25.22	0.00s	0.01s
		Meta-Reflexion	81.65	42.90	10.08	44.98	39.64	25.10	3.01s	0.01s
		ACE	88.18	24.35	6.90	40.58	32.56	25.34	0.00s	0.00s
		MemRL	51.71	15.58	44.77	38.44	36.11	31.36	0.00s	0.00s
		JitRL	**70.87**	**28.54**	**35.14**	**45.66**	**43.62**	**28.34**	**0.00s**	**0.00s**
		Ours	**68.58**	**35.20**	**37.25**	**47.60**	**46.09**	**30.78**	**0.03s**	**0.02s**
	3-8B	No Memory	20.43	10.55	82.11	39.05	31.95	-	-	-
		RAG	23.10	72.47	39.11	43.51	42.10	25.49	0.01s	0.01s
		MEM0	22.59	74.21	40.43	44.32	42.91	28.49	0.01s	5.39s
		MemoryBank	20.10	70.50	38.14	41.53	40.90	29.36	0.01s	7.45s
		A-MEM	22.40	72.38	35.62	42.02	41.05	20.19	1.01s	7.00s
		Reflexion	40.03	81.27	16.05	44.01	41.34	25.36	0.01s	0.01s
		Meta-Reflexion	30.18	69.89	37.56	44.68	44.16	25.04	9.14s	0.00s
		ACE	43.41	67.66	25.64	44.46	43.93	25.00	0.00s	0.00s
		MemRL	30.83	70.59	25.12	40.76	35.77	30.47	0.00s	0.00s
		JitRL	**68.94**	**33.98**	**38.79**	**47.90**	**46.57**	**28.16**	**0.00s**	**0.00s**
		Ours	**70.03**	**40.86**	**45.32**	**52.63**	**50.62**	**30.78**	**0.05s**	**0.04s**
Llama	3.2-3B	No Memory	98.98	3.57	2.96	36.75	19.59	-	-	-
		RAG	93.88	26.33	1.64	41.33	29.87	25.14	0.00s	0.01s
		MEM0	83.69	40.45	3.94	42.81	31.09	26.76	0.01s	1.6s
		MemoryBank	85.44	35.38	4.87	42.22	30.50	27.92	0.00s	1.5s
		A-MEM	85.87	38.15	1.56	42.05	30.32	16.97	1.70s	3.08s
		Reflexion	87.74	48.50	0.43	45.41	36.17	25.34	0.00s	0.00s
		Meta-Reflexion	93.11	31.83	1.25	42.57	31.83	25.10	2.00s	0.01s
		ACE	92.15	37.85	2.56	44.50	33.66	25.23	0.00s	0.00s
		MemRL	78.37	30.16	3.44	37.68	27.52	29.71	0.00s	0.00s
		JitRL	**69.00**	**33.05**	**34.38**	**46.10**	**44.52**	**28.31**	**0.00s**	**0.00s**
		Ours	**70.29**	**39.89**	**36.58**	**49.37**	**47.84**	**29.90**	**0.06s**	**0.07s**
	3-8B	No Memory	98.87	9.97	0	37.60	23.39	-	-	-
		RAG	44.15	82.44	1.23	40.62	34.50	25.08	0.01s	0.01s
		MEM0	39.81	86.81	1.49	40.50	34.10	21.26	0.01s	3.76s
		MemoryBank	46.22	88.37	3.60	43.95	35.71	29.13	0.01s	7.23s
		A-MEM	42.90	80.93	2.19	40.06	34.03	17.73	0.98s	6.87s
		Reflexion	66.33	74.70	1.29	46.08	36.93	24.95	0.00s	0.00s
		Meta-Reflexion	74.49	60.71	2.34	45.10	35.50	24.96	4.12s	0.01s
		ACE	53.06	84.02	1.69	44.37	35.43	24.58	0.00s	0.00s
		MemRL	50.56	87.21	2.33	44.67	35.28	26.40	0.00s	0.00s
		JitRL	**63.06**	**40.17**	**40.30**	**48.23**	**48.11**	**28.24**	**0.00s**	**0.00s**
		Ours	**65.17**	**45.48**	**39.98**	**50.45**	**51.02**	**30.76**	**0.10s**	**0.09s**

producing correspondingly consistent gains. Overall, AuditEvo reveals that the main challenge for current memory agents is not merely preserving past experience, but selectively reusing it when its validity changes across interacting organizational, policy, and risk environments. The error analysis in appendix A.6 provides on further analysis on the impact of environment evolution.

5 Enhance Environment Awareness for Experience Reuse

We propose **Environment-Aware JitRL**, which extends JitRL by introducing an evolving playbook to adapt historical experience retrieval under changing environmental conditions. While JitRL estimates action values from experiences retrieved using predefined state-similarity criteria, our framework continually updates the criteria used to determine which historical experiences are relevant to the current decision. Specifically, the agent maintains an evolving playbook $\mathcal{P}_t$ containing conditional lessons learned from previously resolved cases and their feedback. Given the current case, it selects applicable lessons $\mathcal{P}_t^* \subseteq \mathcal{P}_t$, which guide the current representation mapping ϕ_t. These lessons can introduce, revise, or deactivate features describing observable transaction attributes, relationships,

and temporal patterns. The current case and historical cases are represented under the same current mapping, enabling their similarity to reflect the agent's evolving understanding of relevant decision conditions:

$$\begin{gathered}\mathcal{P}_t^* = \mathrm{Select}(\mathcal{P}_t, x_t, \mathcal{H}_{<t}), \\ s_t = \phi_t(x_t, \mathcal{H}_{<t}), \quad s_i^{(t)} = \phi_t(x_i, \mathcal{H}_{<i}), \quad w_{t,i} = K_{\mathcal{P}_t^*}(s_t, s_i^{(t)}), \quad \mathcal{N}_t = \mathrm{TopK}_{i<t}\, w_{t,i}.\end{gathered} \tag{4}$$

Here, $\mathcal{H}_{<i}$ denotes the history available before historical decision i, and $s_i^{(t)}$ denotes that case's representation under the mapping used at step t. Historical records are retained, allowing their representations to be reconstructed as the playbook evolves. The method therefore adapts both the features used to compare experiences and their relative importance. For example, feedback from repeated purchases involving related vendors may induce a feature capturing cross-vendor transaction patterns that was absent from the earlier representation. These features are derived from observable evidence without supplying the benchmark's latent scenario labels. Following JitRL, the agent estimates action values and advantages from the retrieved state–action–return experiences:

$$\widehat{Q}_t(s_t, a) = \frac{\sum_{i\in\mathcal{N}_t} w_{t,i}\mathbb{I}(a_i = a)G_i}{\sum_{i\in\mathcal{N}_t} w_{t,i}\mathbb{I}(a_i = a)}, \quad \widehat{V}_t(s_t) = \frac{\sum_{i\in\mathcal{N}_t} w_{t,i}G_i}{\sum_{i\in\mathcal{N}_t} w_{t,i}}, \quad \widehat{A}_t(s_t, a) = \widehat{Q}_t(s_t, a) - \widehat{V}_t(s_t). \tag{5}$$

where G_i denotes the observed return. These estimates are computed only when sufficient compatible historical support is available; otherwise, the agent reduces its reliance on memory-based estimates rather than assigning unsupported action values. The estimated advantages adjust the frozen language model's action preferences according to

$$\pi_t(a \mid s_t) = \frac{\pi_\theta(a \mid s_t, \mathcal{P}_t^*)\exp\left(\beta\widehat{A}_t(s_t, a)\right)}{\sum_{a'\in\mathcal{A}_t}\pi_\theta(a' \mid s_t, \mathcal{P}_t^*)\exp\left(\beta\widehat{A}_t(s_t, a')\right)}, \tag{6}$$

where $\mathcal{A}_t$ denotes the current candidate-action set and β controls the influence of historical experience.

After the prediction is finalized and feedback becomes available, an LLM critic examines the current case, retrieved experiences, selected action, and observed reward. It identifies relationships or temporal patterns that the current representation failed to capture and generates conditional lessons for subsequent decisions. The curation procedure incorporates these lessons by introducing new features, revising existing features, or deactivating outdated ones. The playbook is updated through

$$\mathcal{P}_{t+1} = \mathrm{Curate}\left(\mathcal{P}_t, \mathrm{Critic}_{\mathrm{LLM}}\left(x_t, s_t, a_t, r_t, \mathcal{N}_t\right)\right). \tag{7}$$

By coupling feedback-driven representation evolution with context-dependent similarity and experience-based action-value estimation, Environment-Aware JitRL adapts how historical cases are represented, retrieved, and reused as environmental conditions change. The underlying language model remains fixed, while historical case records, original predictions, and observed rewards are retained for subsequent reuse. Implementation details are provided in Appendix A.7, and the adaptation of JitRL to the audit setting is described in Appendix A.8. Table 2 reports accuracy improvements of approximately 2–7 percentage points over JitRL across the evaluated backbones.

6 Conclusion

We introduced **AuditEvo**, a benchmark for evaluating LLM agents under continuously evolving organizational environments. Our experiments show that current memory and reflection frameworks remain unreliable under such evolution, particularly for risk-regime cases. To address this limitation, we proposed Environment-Aware JitRL, which uses immediate feedback to evolve a playbook that conditions state similarity and historical experience retrieval while keeping the underlying LLM fixed. Its consistent improvements over JitRL highlight the importance of adapting not only what experience is retained, but also how past experience is selected and reused under changing conditions. AuditEvo provides a controlled testbed for studying continual memory adaptation, negative transfer, and experience reuse in dynamic organizational decision environments. However, evaluated agents may encounter more diverse or ambiguous evidence patterns in real procurement audits than the benchmark's parameterized scenario families capture.

REFERENCES

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

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Antoine Bigeard, Langston Nashold, Rayan Krishnan, and Shirley Wu. Finance agent benchmark: Benchmarking llms on real-world financial research tasks. *arXiv preprint arXiv:2508.00828*, 2025.

Léo Boisvert, Megh Thakkar, Maxime Gasse, Massimo Caccia, Thibault L De Chezelles, Quentin Cappart, Nicolas Chapados, Alexandre Lacoste, and Alexandre Drouin. Workarena++: Towards compositional planning and reasoning-based common knowledge work tasks. *Advances in Neural Information Processing Systems*, 37:5996–6051, 2024.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

Yanxu Chen, Zijun Yao, Yantao Liu, Amy Xin, Jin Ye, Jianing Yu, Lei Hou, and Juanzi Li. Stockbench: Can llm agents trade stocks profitably in real-world markets? *arXiv preprint arXiv:2510.02209*, 2025.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Chanyeol Choi, Jihoon Kwon, Alejandro Lopez-Lira, Chaewoon Kim, Minjae Kim, Juneha Hwang, Jaeseon Ha, Hojun Choi, Suyeol Yun, Yongjin Kim, et al. Finagentbench: A benchmark dataset for agentic retrieval in financial question answering. In *Proceedings of the 6th ACM International Conference on AI in Finance*, pp. 632–637, 2025.

Romain Froger, Pierre Andrews, Matteo Bettini, Amar Budhiraja, Ricardo Cabral, Virginie Do, Emilien Garreau, Jean-Baptiste Gaya, Hugo Laurençon, Maxime Lecanu, et al. Gaia2: Benchmarking llm agents on dynamic and asynchronous environments. In *International Conference on Learning Representations*, volume 2026, pp. 119758–119789, 2026.

Wenliang Huang, Junkai Li, Zengyi Yu, and Xiangjie Kong. Risctrade: A dynamic multi-market benchmark for risk-mandate compliance of large language models in closed-loop trading. In *International Conference on Intelligent Computing*, pp. 360–370. Springer, 2026.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. Financebench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*, 2023.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Haohang Li, Yupeng Cao, Yangyang Yu, Shashidhar Reddy Javaji, Zhiyang Deng, Yueru He, Yuechen Jiang, Zining Zhu, Kp Subbalakshmi, Jimin Huang, et al. Investorbench: A benchmark for financial decision-making tasks with llm-based agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2509–2525, 2025.

Kenan Li, Rongzhi Li, Linghao Zhang, Qirui Jin, Liao Zhu, Xiaosong Huang, Geng Zhang, Yikai Zhang, Shilin He, Chengxing Xie, et al. Repolaunch: Automating build&test pipeline of code repositories on any language and any platform. *arXiv e-prints*, pp. arXiv–2603, 2026a.

Shuyue Stella Li, Bhargavi Paranjape, Kerem Oktar, Zhongyao Ma, Gelin Zhou, Lin Guan, Na Zhang, Sem Park, Lin Chen, Diyi Yang, et al. Horizonbench: Long-horizon personalization with evolving preferences. *arXiv preprint arXiv:2604.17283*, 2026b.

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. *arXiv preprint arXiv:2601.18510*, 2026c.

Shu Liu, Shangqing Zhao, Chenghao Jia, Xinlin Zhuang, Zhaoguang Long, Jie Zhou, Aimin Zhou, Man Lan, and Yang Chong. Findabench: Benchmarking financial data analysis ability of large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 710–725, 2025.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Mike Merrill, Alexander Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Shin, Thomas Walshe, E Kelly Buchanan, et al. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *International Conference on Learning Representations*, volume 2026, pp. 40903–40986, 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 arXiv:2310.08560*, 2023.

N Shinn, F Cassano, E Berman, A Gopinath, K Narasimhan, and SR Yao. Language agents with verbal reinforcement learning. *arXiv*, 2023.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Tianxin Wei, Noveen Sachdeva, Benjamin Coleman, Zhankui He, Yuanchen Bei, Xuying Ning, Mengting Ai, Yunzhe Li, Jingrui He, Ed H Chi, et al. Evo-memory: Benchmarking llm agent test-time learning with self-evolving memory. *arXiv preprint arXiv:2511.20857*, 2025.

Chunlong Wu, Ye Luo, Zhibo Qu, and Min Wang. Meta-policy reflexion: Reusable reflective memory and rule admissibility for resource-efficient llm agent. *arXiv preprint arXiv:2509.03990*, 2025.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, et al. Finben: A holistic financial benchmark for large language models. *Advances in neural information processing systems*, 37:95716–95743, 2024.

Jundong Xu, Qingchuan Li, Jiaying Wu, Yihuai Lan, Shuyue Stella Li, Huichi Zhou, Bowen Jiang, Lei Wang, Jun Wang, Anh Tuan Luu, et al. Evoarena: Tracking memory evolution for robust llm agents in dynamic environments. *arXiv preprint arXiv:2606.13681*, 2026a.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026b.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. Augmentation-adapted retriever improves generalization of language models as generic plug-in. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2421–2436, 2023.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, volume 2026, pp. 86069–86100, 2026a.

Shengtao Zhang, Jiaqian Wang, Ruiwen Zhou, Junwei Liao, Yuchen Feng, Zhuo Li, Yujie Zheng, Weinan Zhang, Ying Wen, Zhiyu Li, et al. Memrl: Self-evolving agents via runtime reinforcement learning on episodic memory. *arXiv preprint arXiv:2601.03192*, 2026b.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19724–19731, 2024.

A Appendix

A.1 Ensuring High-Quality and Leakage-Resistant Data

AuditEvo incorporates a deterministic validation layer after case generation and oracle evaluation. Because the ground truth is produced by the simulator itself, quality assurance focuses on internal logical consistency, causal validity, scenario realization, and separation between observable and oracle-only information, rather than relying on an LLM-as-a-judge. First, each case is checked for basic transactional validity. The validator verifies that the attached policy corresponds to the policy effective on the request date, request and invoice amounts are positive, realized invoice amounts remain within the intended tolerance of request amounts, and procurement lifecycle dates satisfy required ordering constraints. It additionally checks that normal payment observations conform to their department-specific baseline profile and that the oracle's cumulative-spending arithmetic is internally consistent. Second, the benchmark validates label consistency. The generated audit opinion must agree exactly with the deterministic precedence rule over policy violations and risk signals. The simulator also prevents targeted risk cases from inadvertently combining multiple fraud families, thereby making scenario-level evaluation interpretable. Third, each deliberately generated scenario is checked for successful realization. Clean anchors must remain clean, policy-target cases must produce the intended policy violation without an accompanying fraud finding, and risk-target cases must produce an 'investigation' label without policy violations. Specific assertions verify the realization of split-purchase, price-inflation, late-delivery, premature-invoice, payment-timing, concealed-ownership, emergency-abuse, procurement-card, sole-source-concentration, and bid-rotation scenarios. These checks are important because the mere intention to generate a difficult scenario does not guarantee that its downstream interaction with policy, prior cases, and other simulator variables yields the desired evaluation condition. Finally, AuditEvo is explicitly designed against temporal and annotation leakage. The evaluated agent is restricted to history preceding the current case; canonical vendor IDs, beneficial-ownership groups, latent scenario types, environment patches, and oracle rationale remain outside the primary public input, and unlabeled history views remove retrospective audit outcomes when the experimental setting assumes no historical feedback. The generator additionally maintains separate public and gold artifacts and records aggregate validation statistics in a manifest, facilitating reproducible inspection of generated benchmark releases.

A.2 Source-Case Attribution and Reference-Set Diagnostics

Source-case identifiers are obtained through strategy-specific memory adapters. For A-MEM, the adapter reconstructs identifiers from selected notes and links separately from the rendered memory text. These identifiers therefore represent adapter-reconstructed source attribution rather than an exact native retrieval log. For summarized memories, retrieved content is mapped to source cases where provenance is available; for example, the Mem0 adapter uses retained source-event metadata. Such attribution identifies source associations but does not verify whether the retrieved representation preserves the decision-relevant facts from those cases. Reference-set coverage also affects interpretation. Because the oracle-reference set is not exhaustive, an alternative group of valid market peers can be counted as false positives despite supporting a correct assessment. When the reference set is empty, any returned identifiers contribute false positives under the overlap metric, whereas returning no identifiers contributes no true-positive, false-positive, or false-negative counts. An empty reference set is not interpreted as evidence that historical information cannot help the decision.

A.3 Benchmark Validation

We assess public-evidence validity through three complementary tests. First, a text round-trip audit parses every agent-visible case description and policy document and compares decision-relevant amounts, dates, vendor attributes, procurement fields, and policy requirements with their structured source values. This audit checks information preservation while confirming that hidden simulator fields and oracle-only annotations are not directly exposed. Second, an independently implemented public-only oracle reconstructs findings and audit labels using only the current case, its attached policy, chronologically eligible historical cases, and their policy attachments and feedback. Hidden scenario annotations, organizational state, and generator functions are unavailable during reconstruction; the gold annotation file is accessed only afterward for scoring. Third, a calibration-identifiability analysis enumerates plausible detector parameters consistent with the labeled calibration cases. Within the specified rule family and parameter ranges, a scored task is classified as identifiable only when the consistent-parameter set is nonempty and every member produces the same audit label. We report field-level fidelity, public-oracle agreement, the proportion of identifiable tasks, and ambiguity by finding family in table 3, thereby distinguishing preservation of observable information from identification of the underlying decision rules.

To reduce shortcuts arising from the generation schedule, scenario placement is randomized separately within each organizational chain and generation cycle while preserving required temporal dependencies. For example, the anchor transactions supporting a split-purchase finding must precede its target, but the complete episode can occupy different positions across chains and cycles. This design reduces systematic associations between predefined scenario slots and audit outcomes while retaining the historical evidence required for multi-case reasoning. Schedule randomization is treated as a construction safeguard rather than a guarantee that all statistical shortcuts have been eliminated.

Automated validation is supplemented by review from practicing or former professional auditors. For each case, we assign three auditors to review. The auditors review the oracle rules and, for every generated case, its structured representation and naturalized description. This review assesses the plausibility of the audit logic, the consistency of the generated case with that logic, and the fidelity of its textual presentation. Expert assessment therefore complements the automated reconstruction and consistency checks by examining whether the benchmark's formal conditions are represented as coherent procurement-audit cases.

We define the audit time of case i as its closing date, $t_i = \mathrm{close_date}_i$, and admit a transaction into resolved historical memory only after it closes. Accordingly, the eligible history is $\mathcal{H}_i = \{j : \mathrm{close_date}_j < \mathrm{close_date}_i\}$. The current case contains its complete request-through-payment lifecycle, and the generator enforces $\mathrm{payment_date}_i \leq \mathrm{close_date}_i$. Within the eligible evidence, policy selection uses the request date, spending aggregation uses the invoice date, and risk detectors apply their corresponding order, invoice, delivery, or payment dates. An earlier transaction that has not yet closed is unavailable as resolved historical evidence. This separates the time at which evidence becomes available from the event dates used to interpret it and establishes the task as retrospective auditing of completed transactions.

A.4 Dataset Statistics

Table 4 displays descriptive statistics for AuditEvo.

A.5 Task Prompt

Figure 1 displays task prompt.

A.6 Error Analysis

To assess how environmental evolution contributes to audit errors, we compare the original AuditEvo setting with a matched control in which procurement policies, market-price distributions, and vendor and risk-regime parameters remain fixed within each organizational chain. The control preserves organizational heterogeneity, chronological case arrival, historical dependencies, and the availability of verified feedback after each prediction. Risky transactions and multi-case patterns remain present, ensuring that the comparison retains the underlying audit tasks. We generate the two settings using

Table 3: Validation of public-evidence fidelity, label reconstructibility, calibration identifiability, and expert review. Fidelity and reconstruction results refer to the final dataset release; strict identifiability and expert-review results are reported only where independently established.

Validation measure	**Result**
A. Public-text fidelity	
Cases checked	43,100
Policy documents checked	500
Decision-relevant fields checked	3,121,000
Correctly preserved fields, count (%)	3,121,000 (100.0%)
Cases with all required fields preserved, count (%)	43,100 (100.0%)
Cases exposing prohibited simulator fields	0 (0.0%)
B. Public-only reconstruction	
Scored cases evaluated	40,000
Exact audit-label agreement, all scored cases (%)	40,000 (100.0%)
Exact audit-label agreement, identifiable cases (%)	40,000 (100.0%)
Abstentions or reconstruction failures, count (%)	0 (0.0%)
C. Calibration identifiability	
Identifiable scored tasks, count (%)	40,000 (100.0%)
Ambiguous scored tasks, count (%)	0 (0.0%)
Tasks with no consistent parameter setting, count (%)	0 (0.0%)

Task Prompt

```
Task: [case description]
Determine the retrospective audit opinion for this case based on
    the purchasing policy attached to this case and any relevant
    resolved prior procurement cases available. Decision rule:
    return "investigation" if the case has any potential risk
    finding; return "reject" if the case has only policy violation;
    return "investigation" if you think the case contains both risk
    finding and policy violation. Otherwise, return "approve".
    Return only one JSON object matching the schema below. In
    Reasoning, provide a complete explanation of how the current
    case facts, applicable policy, and relevant historical evidence
    support the selected decision. Include all material findings and
     calculations, but do not provide hidden chain-of-thought,
    unrelated analysis, or additional text.

{
  "audit_opinion": "approve | investigation | reject",
  "reasoning": "complete explanation supporting the decision"
}
```

Figure 1: Task prompt used for procurement audit evaluation.

matched initial organizational configurations and random seeds, maintaining comparable chain lengths, class proportions, and scenario composition, and recompute historical evidence and oracle labels under the corresponding environmental conditions. We evaluate a representative subset of LLM backbones (GPT-5, Qwen3-8B) and memory frameworks with identical prompts, memory budgets, and update procedures across settings. We report audit-opinion accuracy and macro-F1 to quantify performance degradation associated with environmental evolution. We additionally compare the performance advantage of Environment-Aware JitRL over JitRL between the fixed and evolving settings. A larger advantage under evolution would support the interpretation that the proposed method is particularly beneficial when environmental conditions change, whereas similar gains across settings would suggest a more general improvement in experience reuse. This comparison helps distinguish the additional challenge introduced by environmental evolution from

Table 4: Distribution of the 40,000-Case AuditEvo Dataset

Panel	Category	Cases	Percentage
A. Dataset scale			
	Independent organizational chains	100	–
	Scored cases per chain	400	–
	Total scored cases	40,000	100.0%
	Unscored calibration cases	3,100	–
	Total historical memory events	43,100	–
	Balanced schedule cycles	1,000	–
B. Mutually exclusive finding categories			
	Clean cases	14,000	35.0%
	Policy-violation-only cases	12,000	30.0%
	Risk-only cases	14,000	35.0%
	Cases containing both policy and risk findings	0	0.0%
C. Retrospective audit-opinion labels			
	Approve	14,000	35.0%
	Investigation	14,000	35.0%
	Reject	12,000	30.0%
D. Policy-violation-only cases			
	Incorrect competitive-bid process	5,000	12.5%
	Incorrect high-value purchasing method	3,000	7.5%
	Insufficient quotations	2,000	5.0%
	Vendor–department fiscal-year threshold violation	2,000	5.0%
	Subtotal	*12,000*	*30.0%*
E. Risk-only cases			
	Payment-timing anomaly	3,000	7.5%
	Sole-source/vendor-concentration abuse	2,000	5.0%
	Split purchasing	2,000	5.0%
	Bid-rotation collusion	1,000	2.5%
	Concealed ownership	1,000	2.5%
	Emergency-purchase abuse	1,000	2.5%
	Premature invoicing	1,000	2.5%
	Late delivery	1,000	2.5%
	Vendor-specific price inflation	1,000	2.5%
	Procurement-card misuse	1,000	2.5%
	Subtotal	*14,000*	*35.0%*
F. Unscored calibration memory			
	Clean calibration and anchor cases	2,000	64.5%[a]
	Investigation exemplars	1,100	35.5%[a]
	Total calibration cases	*3,100*	*100.0%*[a]

[a] Calibration percentages use 3,100 unscored calibration cases as the denominator. All other percentages use 40,000 scored cases.

Payment-timing anomalies contain approximately 1,500 unusually early and 1,500 unusually late payment cases.

Finding categories are mutually exclusive. Consequently, each risk-only case receives an `investigation` label, each policy-violation-only case receives a `reject` label, and each clean case receives an `approve` label.

Operational procurement approval status is a separate case attribute and is not represented by the retrospective audit-opinion distribution.

Table 5: Performance under fixed environmental conditions. Accuracy and macro-F1 are reported as percentages. Chronological case arrival, historical dependencies, and online feedback are preserved.

Model	Framework	Normal	Policy-Violation	Risk-Regime	Acc.	Macro-F1
GPT-5	RAG	80.36	78.99	25.44	60.73	58.15
	ACE	65.58	70.12	48.77	61.06	60.22
	JitRL	70.79	52.45	48.23	57.39	56.49
	Ours	72.15	60.70	49.81	60.90	59.03
Qwen3-8B	RAG	58.19	60.38	40.20	52.55	50.30
	ACE	57.35	68.55	39.00	54.29	51.75
	JitRL	70.81	44.28	47.05	54.54	52.67
	Ours	70.44	47.60	47.64	55.61	53.88

the benefits of accumulating labeled examples and learning stable organizational conventions. The results demonstrated in table 5 suggest the advantage gap narrows under static environment.

A.7 Implementation Details for Environment-Aware JitRL

We intentionally avoid defining the state representation using a large set of fraud-specific dimensions to ensure a fair comparison with other frameworks, since doing so would provide the agent with prior knowledge of the latent risk regimes that AuditEvo is designed to require it to discover. Instead, the initial state $s_t^{(0)} = \phi_0(x_t, H_{<t})$ is constructed only from generic observable information available in the procurement workflow, including the current transaction, vendor and department identifiers, amounts and dates, applicable policy text, and chronologically available prior cases. Historical cases are retained in their raw form together with their actions and observed feedback, rather than being permanently compressed into a fixed state vector. The evolving playbook $\mathcal{P}_t$ is then used to expand the representation online. After each prediction, immediate feedback is provided and an LLM critic examines the current case, retrieved experiences, decision, and outcome to determine whether the existing representation failed to capture a reusable relationship or temporal pattern. A newly generated lesson ℓ_k may therefore induce a new retrieval feature f_k, revise an existing feature, deactivate an outdated feature, or leave the representation unchanged. The representation available at time t consequently takes the form $s_t = [\phi_0(x_t, H_{<t}), f_1(x_t, H_{<t}), \ldots, f_{K_t}(x_t, H_{<t})]$, where K_t grows or changes with accumulated feedback rather than being fixed in advance. Importantly, induced features are expressed in terms of observable relationships, such as repeated transactions across related entities or recurring temporal patterns, rather than benchmark-specific labels such as 'bid rotation" or 'concealed ownership." For retrieval, the currently active playbook defines the evolving mapping ϕ_t used to represent and compare the current case with eligible historical cases. Historical candidate i is therefore reconstructed lazily under the current representation, $s_i^{(t)} = \phi_t(x_i, H_{<i})$, using only information that was chronologically available before case i. Each active lesson produces a nonnegative relevance score for its induced feature, which is normalized across the currently active features to obtain weights $\alpha_{t,k}$, and the playbook-conditioned similarity is computed as $K_{\mathcal{P}_t^*}(s_t, s_i^{(t)}) = \sum_{k=0}^{K_t} \alpha_{t,k} K_k(s_t^{(k)}, s_i^{(k,t)})$. Relevant lessons $\mathcal{P}_t^*$ are selected according to their applicability to the current case and recent observable context, while conflicting lessons are revised or deactivated using subsequent feedback rather than accumulated as independent rules. This design allows the agent to discover new dimensions of experience reuse during evaluation without being given the benchmark's latent fraud taxonomy.

A.8 The adaptation of JitRL to AuditEvo

JitRL was originally formulated for multi-step agentic trajectories, where an action denotes a sub-decision within a longer interaction (e.g., a tool call or navigation step) and reward is typically LLM-judged in the absence of ground truth. AuditEvo differs from this setting in two respects that require an explicit adaptation. First, each task is a single terminal classification rather than a multi-step trajectory: the agent observes one case and its available policy and history, then emits one audit opinion. We therefore define the action space at each decision as the three-way audit label itself, $A_t = \{\text{approve}, \text{reject}, \text{investigation}\}$, rather than a sub-step within an extended rollout. Second,

AuditEvo's retrospective protocol reveals a deterministic, verified outcome immediately after each prediction. Thus, we use this ground-truth signal directly as reward rather than substituting an LLM-based judge, consistent with the benchmark's broader design choice to avoid LLM-based annotation of case outcomes. Concretely, memory stores one record per resolved case, $M_t = \{(s_i, a_i, G_i)\}_{i<t}$ where $a_i = \hat{y}_i$ is the agent's own emitted prediction at step i, *not* the oracle label y_i, and $G_i = 2 \cdot \mathbf{1}[\hat{y}_i = y_i] - 1 \in \{-1, +1\}$ is the realized correctness reward disclosed by the environment immediately after prediction. Storing the agent's own action rather than the corrected label is essential to the method's validity: if memory instead recorded the oracle label as the action for every case, every stored example would be "correct by construction," collapsing $\hat{Q}_t(s_t, a)$ and $\hat{V}_t(s_t)$ to the same constant for any retrieved neighborhood and eliminating the advantage signal. We retain both correct and incorrect past predictions in memory without filtering. Retaining failures is necessary for the same reason and is additionally central to the method's purpose: a cluster of failed predictions sharing a stale similarity profile is precisely the negative-transfer signal that Environment-Aware JitRL's playbook is designed to detect and downweight, discarding failed examples would remove this signal entirely.

We implement $\pi_\theta(a \mid s_t)$ in using verbalized confidence scoring uniformly across all backbones. The JSON schema in Figure 1 is extended with a required `action_probabilities` field constrained to sum to 1 over the three labels, and the model's self-reported distribution is used directly as $\pi_\theta(a \mid s_t)$. We adopt a single scoring procedure across backbones, rather than token-probability scoring for open-weight models, to hold the policy-estimation procedure fixed across comparisons and to avoid conflating differences in scoring mode with differences in memory strategy or backbone capability. The final prediction at each step is taken as $\hat{y}_t = \arg\max_{a \in A_t} \pi_t(a \mid s_t)$, so that evaluation is deterministic given a fixed memory state and backbone. When the retrieved neighborhood N_t contains no example with $a_i = a$ for a given action, the numerator and denominator of $\hat{Q}_t(s_t, a)$ are both zero, we define $\hat{A}_t(s_t, a) = 0$ in this case, leaving $\pi_\theta(a \mid s_t, P_t^*)$ unmodified for that action rather than assigning it an undefined value. If no action has support from at least $k_{\min} = 3$ retrieved examples, the method forgoes memory-based correction entirely for that step and returns the uncorrected base policy $\pi_\theta(a \mid s_t, P_t^*)$.

A.9 Declaration of LLM Usage

We used LLMs to assist with writing, polishing, and synthetic data generation. All technical ideas, experimental design, analysis, and final manuscript decisions were made by the authors.