CONTINUAL LEARNING, NOT TRAINING: ONLINE ADAPTATION FOR AGENTS

Aman Jaglan[1] and Jarrod Barnes[1]

[1]*Arc Intelligence* , {aman, jarrod}@arc.computer
Corresponding author: jarrod@arc.computer

October 28, 2025

Abstract

Continual Learning (CL) methods have traditionally focused on mitigating catastrophic forgetting through gradient-based retraining, an approach ill-suited for deployed agents that must adapt in real time. We introduce our Adaptive Teaching and Learning System (ATLAS), a dual-agent architecture that decouples reasoning (Teacher) from execution (Student) and incorporates a persistent learning memory that stores distilled guidance from experience. This informs the orchestration layer, enabling the system to dynamically adjust its operational strategies, such as supervision level or initial plan selection, at inference time. In doing so, ATLAS achieves gradient-free continual learning, shifting the locus of adaptation from model parameters to system-level orchestration. We formulate this as a system-centric paradigm for continual learning, where the objective is adaptive efficiency: maximizing task success while minimizing computational cost through inference-time orchestration rather than parameter updates (Kirkpatrick et al., 2017; Rolnick et al., 2019). Evaluated on Microsoft's ExCyTIn-Bench (Incident #5 subset) (Wu et al., 2025), an open-source benchmark simulating complex cyber-threat investigation, ATLAS achieves 54.1% success with GPT-5-mini as its Student, outperforming the larger GPT-5 (High) by 13% while reducing cost by 86%. Inference-time continual learning positions this approach on the Pareto frontier: superior accuracy at lower computational cost, achieved through gradient-free adaptation during deployment. The system demonstrates progressive efficiency gains across the 98-task trajectory, reducing token consumption from 100,810 (tasks 1–25) to 67,002 (tasks 61–98) while maintaining mid-50% success, confirming that ATLAS learns to solve incidents more economically without sacrificing accuracy. Cross-incident validation on Incident #55 demonstrates generalization: frozen pamphlets from Incident #5 improve accuracy from 28% to 41% (+46%) with zero retraining, while shifting output composition from verbose exploration to structured reasoning (−52% non-reasoning tokens, +2,135 reasoning tokens per question). Together, these findings establish gradient-free continual learning as a viable path toward adaptive, deployable AI systems and provide causally annotated traces valuable for training explicit world models (Yu et al., 2023). Code and reproducibility assets are available at `https://github.com/Arc-Computer/atlas-sdk`; the accompanying dataset will be released to enable replication and further study.

***K*eywords** Continual Learning, Agent Architecture, Inference-Time Adaptation, LLM, Gradient-Free Learning

Acknowledgments

The authors would like to thank **Gabriella Haffner** and **Michelangelo Naim** for their valuable contributions, discussions, and feedback throughout the development of this work.

1 Introduction

Deployed language model agents operate in dynamic environments requiring continuous adaptation, yet their core knowledge remains static after pretraining (Pham et al., 2021). This creates a tension, how can systems adapt in real-time when the dominant learning paradigms rely on offline training cycles? While Continual Learning (CL) addresses knowledge updates, existing approaches focus overwhelmingly on mitigating catastrophic forgetting through gradient-based weight updates conducted offline (Kirkpatrick et al., 2017; Rolnick et al., 2019), an inherently model-centric paradigm ill-suited for deployment constraints.

In complex adaptive systems, the environment perpetually evolves, by the time a model completes offline training on one configuration, the live system may have already shifted. Backpropagation, even in efficient forms like parameter-efficient fine-tuning (PEFT, e.g., LoRA; Hu et al. 2021) or sparse-update methods, necessitates dedicated training loops, specialized hardware, data accumulation, and introduces retraining delays. These approaches cannot provide the inference-time adaptation needed for agents operating in real-time, often on resource-constrained hardware or without access to training infrastructure (Guo et al., 2021; Dettmers et al., 2023; Liu et al., 2024; Frantar & Alistarh, 2023).

To address this challenge, we propose a system-centric approach to continual learning designed for inference-time deployment. Rather than focusing on updating weights without forgetting, we reframe the goal as achieving efficient performance: measurable improvements in task success and reductions in computational cost e.g., tokens consumed, as the system gains experience during live operation. This requires shifting the locus of adaptation from model weights to the system's orchestration layer.

We introduce ATLAS, a dual-agent architecture operationalizing this paradigm shift from model- to system-centric continual learning. ATLAS achieves gradient-free adaptation at inference-time through memory-guided orchestration, using aggregated learning history and rewards derived from Teacher-Student interactions to dynamically adjust its operational strategy. This mechanism requires no gradients, no model retraining, and no specialized hardware, making adaptation immediate, cost-efficient, transparent, and deployable by practitioners on standard inference hardware.

On Microsoft's ExCyTIn-Bench (Incident #5), ATLAS lifts success of GPT-5-mini from 33.7% to 54.1%, trims the Student's tokens from 141,660 to 78,118 (−45%), and surpasses the larger GPT-5 (High) by 13%. This system-centric process serves as a novel data engine, demonstrating how its adaptive mechanism naturally generates the causally-annotated traces needed for training explicit world models (Yu et al., 2023; Ha & Schmidhuber, 2018).

2 Related Work

The canon of current literature relating to CL falls under four main categories: (1) training-based approaches that suffer from catastrophic forgetting and require computationally expensive gradient updates (Kirkpatrick et al., 2017; Rolnick et al., 2019; Pham et al., 2021; Hu et al., 2021; Guo et al., 2021; Dettmers et al., 2023; Liu et al., 2024; Frantar & Alistarh, 2023), (2) prompt optimization techniques that produce static instructions for deployment (Lester et al., 2021; Khattab et al., 2023; Agrawal et al., 2025), (3) retrieval-augmented systems that perform lookup rather than skill synthesis (Lewis et al., 2020; Guu et al., 2020; Borgeaud et al., 2022; Asai et al., 2024; Lin et al., 2024), and (4) agent memory mechanisms that passively store experiences without extracting generalizable knowledge (Shinn et al., 2023; Zhou et al., 2024; Wang et al., 2023; Packer et al., 2023). Importantly none of these methods enable closed-loop, gradient-free skill refinement during deployment. Building on existing work in continual learning, we argue that current approaches fail to support real-time adaptation, as they remain anchored to gradient-based retraining. We instead propose a system-centric framework that enables gradient-free, inference-time adaptation by decoupling reasoning from execution.

2.1 Training-based

The current dominant method in CL is training-based, with a focus on mitigating catastrophic forgetting via updating weights. Methods such as LoRA (Hu et al., 2021), QLoRA (Dettmers et al., 2023), and DoRA (Liu et al., 2024) , sparse-update mechanisms, and experience replay techniques reduce computational costs but remain fundamentally constrained by their reliance on gradient-based optimization (Hu et al., 2021; Guo et al., 2021; Frantar & Alistarh, 2023; Dettmers et al., 2023; Liu et al., 2024). Even recent proposals for "fast-slow" dual-speed learning systems that implement rapid parameter updates still depend on gradient computation and cannot achieve immediate behavioral modification during task execution (Pham et al., 2021).

These learning methods face a tension, while aggressive learning rates precipitate the instability of catastrophic forgetting, overly cautious updates restrict the agent's ability to achieve sufficient generalization and adaptation. Our system-centric approach with ATLAS, offers a departure from this method by treating model parameters as static and moving adaptation from the training loop into the inference-time orchestration of agent interactions, enabling immediate learning without gradient computation or forgetting.

2.2 Prompt Optimization

Representative systems include Prompt Tuning (Lester et al., 2021), DSPy compilation with self-improving pipelines (Khattab et al., 2023), and recent GEPA evolutionary methods that outperform RL-style optimizers in sample efficiency

(Agrawal et al., 2025), while other methods leverage gradient-based prompt tuning or reinforcement learning over discrete prompt spaces. These techniques assess what is the optimal prompt for a task but are static and do not evolve in environmental conditions post-deployment.

ATLAS allows agents to continuously adapt their execution strategy based on a history of task-specific successes and failures. This stateful learning, which emerges dynamically during inference, enables highly context-specific specialization that addresses novel failure modes static prompts cannot handle.

2.3 Retrieval Systems

Retrieval mechanisms such as Retrieval-Augmented Generation (RAG) augment models by retrieving relevant documents or examples from external memory to provide the model with greater context (Lewis et al., 2020; Guu et al., 2020; Borgeaud et al., 2022). More adaptive retrieval training such as Self-RAG and RA-DIT improves when/what to retrieve (Asai et al., 2024; Lin et al., 2024).

ATLAS stores reward trajectories, which include past Teacher feedback, execution outcomes, and the corrective strategies employed. These trajectories are then utilized to train a meta-level control policy, enabling dynamic adjustments, such as varying the level of Teacher supervision based on the Student agent's performance patterns. This distinction shifts learning from the content level, knowledge augmentation to the strategic level, behavioral policy refinement, enabling skill acquisition rather than expanding content.

2.4 Memory Mechanisms

Episodic memory systems record interaction histories to improve future performance. Approaches such as Reflexion (Shinn et al., 2023), LATS (Zhou et al., 2024), Voyager (Wang et al., 2023), and MemGPT (Packer et al., 2023) maintain logs of past actions, outcomes, and textual reflections that agents can reference during subsequent tasks. This line of research implements memory as a passive, episodic log, a chronological record of observations and interactions. These systems lack a mechanism for active compression and generalization, memory grows with experience but past failures are not processed into generalizable knowledge that can steer future actions and alter execution policy. These systems also suffer from unbounded memory growth and lack principled consolidation and accumulate episodic traces without synthesizing higher-order behavioral rules.

In contrast, ATLAS implements memory as an active learning substrate, not a passive log. The Teacher's corrective feedback is explicitly structured and annotated, allowing the orchestrator to use this refined history to directly steer the Student's policy. This transformation enables progressive skill refinement and true procedural learning at inference time, all without modifying the model's underlying parameters.

3 Methodology

To operationalize our system-centric paradigm for continual learning, we designed the ATLAS architecture specifically for gradient-free, inference-time adaptation. Its core components enable dynamic behavioral adjustments based on accumulated experience, directly addressing the goal of efficient performance without modifying underlying model weights. Our methodology centers on a dual-agent architecture engineered to decouple reasoning (Teacher) from execution (Student).

System Architecture: The core of our system is an adaptive runtime orchestrating the interaction between a primary **Student** agent and a typically more capable **Teacher** agent. For each task, this interaction unfolds sequentially:

1. **Task Execution:** The Student attempts the task, generating a trajectory of states, actions (e.g., tool calls, queries), and observations.
2. **Guidance and Verification:** The Teacher observes the Student's trajectory. Based on task outcome and efficiency, it provides corrective, principle-level guidance.
3. **Learning Persistence:** The runtime records the complete execution trace, Teacher guidance, and associated scores (derived from a two-tier reward system providing auditable rationales) in a **Persistent Learning Memory (PLM)**, indexed by task context. A lightweight process distills actionable guidance from these records.

This cycle enables Adaptive Learning at Inference Time. On subsequent similar tasks (identified by task context), the runtime retrieves the relevant aggregated learning history (distilled guidance and past performance) from the PLM. This retrieved information directly informs the orchestration layer, which dynamically adjusts the system's operational

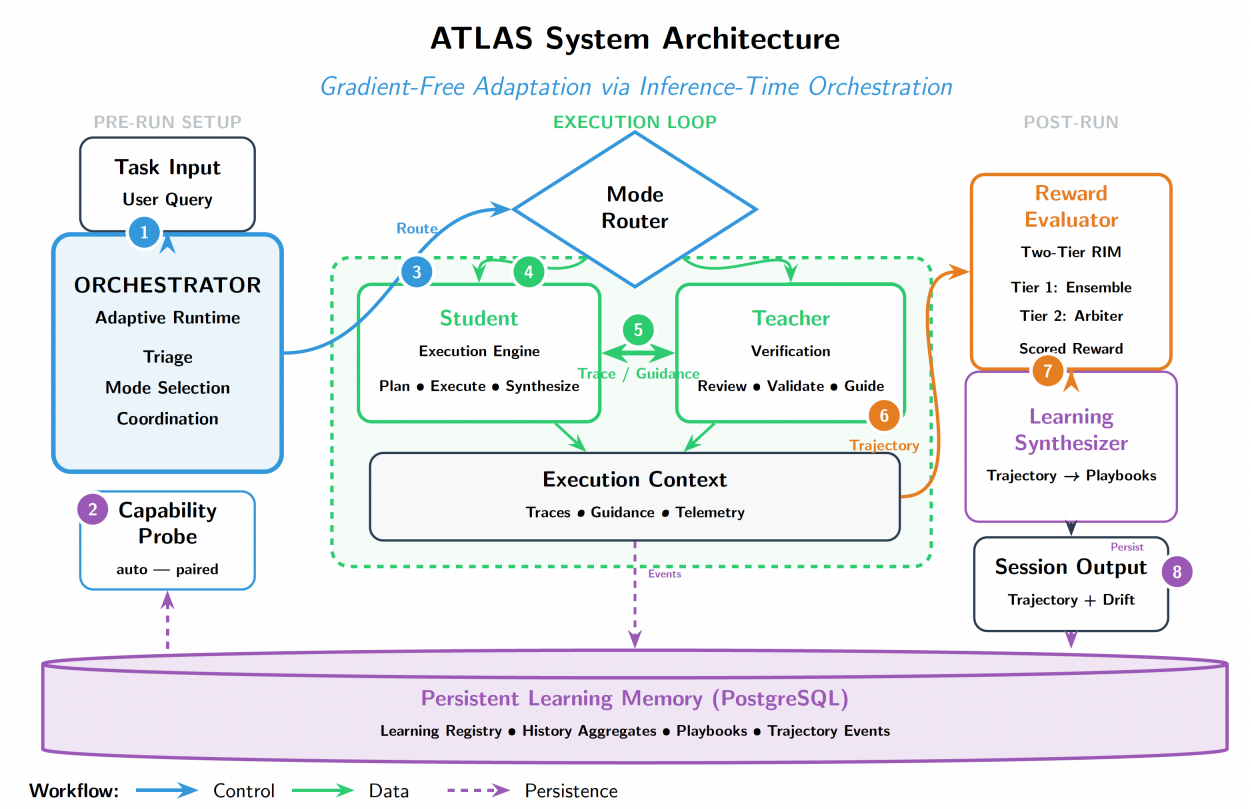

Figure 1: ATLAS System Architecture for gradient-free adaptation. An Orchestrator manages Teacher-Student interactions during the execution loop. Post-run, learning is evaluated, synthesized, and stored in Persistent Learning Memory (PLM) to guide future inference-time decisions.

strategy, for example, selecting the appropriate Teacher supervision level (e.g., fully autonomous for high-confidence tasks, step-by-step guidance for novel ones) or seeding the Student's plan.

This entire adaptation process occurs purely at inference time, leaving model weights unchanged. Learning is mediated solely through the structured history stored in the PLM and retrieved by the orchestrator, enabling the system to develop "fast paths" for familiar scenarios while invoking escalated supervision for novel or challenging ones, thus transforming raw interaction traces into actionable, reusable learning.

3.1 Inference-Time Learning

Adaptation arises from a mechanistic loop. When the Student fails or demonstrates inefficiency, the Teacher provides concise, principle-level guidance, such as verifying the source IP and authentication path before conducting privilege analysis in a security triage task. The reward subsystem then evaluates both the Student's trajectory and the Teacher's intervention along predefined axes including factuality, instruction-following, efficiency, and safety, attaching structured rationales to each score. The learning engine subsequently compiles two artifacts: a Teacher Pamphlet, which captures principles, failure modes, diagnostics, and stop conditions; and a Student Pamphlet, which encodes the corresponding action schema, tool plan, guards, and success checks.

3.2 Reward System

To transform raw interactions into high-fidelity supervision, ATLAS employs a two-tier, ensemble-of-judges rewarder. Multiple fast judges independently score trajectories and guidance, each required to state the evaluation principles before assigning scores; when variance or self-reported uncertainty exceeds thresholds, a stronger arbiter consolidates the rationales and issues the final judgment (Jung et al., 2025) This routing preserves low cost on routine cases while allocating deliberation to ambiguous ones, and the resulting principle-grounded rationales form an audit trail for prompt and threshold tuning. Publicly reported results indicate this system attains high accuracy on RewardBench 2

(Malik et al., 2025) via an ensemble-then-arbiter design, consistent with our use as an auditable reward signal within ATLAS.

3.3 World-Model Data Engine

A direct consequence of the ATLAS learning loop is the generation of structured and causally annotated data suitable for world-model training (Yu et al., 2023). Each execution trace contains three components: (1) state: task context, environment state, and intermediate observations; (2) action: the Student's tool/API/SQL calls, plans, and decision points; and (3) outcome: observations, success and failure signals, artifacts, latencies, and retries. These traces also include the Teacher's diagnostic guidance explaining why particular actions failed or succeeded, along with meta-signals such as supervision-lane decisions, confidence estimates, disagreement, and escalation events. Unlike raw or success-only logs, these traces provide explicit causal explanations for action outcomes and cover both optimal and correction-rich trajectories. We hypothesize that world models trained on ATLAS traces will exhibit improved predictive fidelity and sample efficiency relative to models trained on conventional trajectory data.

4 Experimental Setup

4.1 Benchmark

The experiments are conducted on ExCyTIn-Bench (Wu et al., 2025), a cyber-threat investigation benchmark designed for stateful reasoning. As CL shifts evaluation away from static test sets, ExCyTIn-Bench offers a more process-aware assessment by scoring trajectories within a simulated incident environment. We focus on Incident #5, which provides a consistent scenario and scoring protocol.

4.2 System Configuration and Baselines

System Configuration

Our experimental setup consists of two phases: seeding and evaluation. During the seeding phase, we employ a paired configuration where GPT-5 serves as the Teacher model and GPT-5-mini as the Student model. Working on ExCyTIn-Bench Incident #5 (98 queries), the Teacher observes Student trajectories and provides targeted guidance. These interactions are then distilled into Learning Pamphlets and stored in the Persistent Learning Memory (PLM).

In the evaluation phase, we assess cross-task transfer by retrieving relevant pamphlets via semantic similarity to initialize subsequent tasks. This tests whether guidance learned from earlier queries can improve performance on later, related queries within the same incident domain, without any model weight updates, maintaining the inference-time learning paradigm.

Baseline

We establish two comparison points to isolate the contribution of our approach:

1. Internal baseline (Student-only): GPT-5-mini operating without pamphlet or Teacher guidance. This isolates the impact of our inference-time learning mechanism by measuring raw Student performance.
2. External baseline (Benchmark reference): The reported GPT-5 (Reasoning = High) performance on Incident #5 from ExCyTIn-Bench documentation, which achieved an average reward of 0.501. This provides a reference point from a stronger model using the benchmark's standard evaluation protocol.

All experiments use identical reward configurations and evaluation protocols to ensure standardized comparison across baselines and our system.

4.3 Metrics

We measure performance across two metrics:

1. **Task Success Rate:** binary success rate computed using the benchmark's official criterion, reported both overall and on the benchmark's flagged query subset, as defined by ExCyTIn-Bench. This provides a strict pass or fail assessment of task completion. Success is determined using ExCyTIn-Bench's binary correctness evaluation, which applies a threshold of $\geq$0.4 on the ensemble reward score to account for judge uncertainty while maintaining strict answer verification.

2. **Efficiency:** average tokens consumed per session, measuring the computational cost of achieving task objectives.

5 Results

We evaluate the paired Teacher–Student configuration using ExCyTIn-Bench's scoring protocol on Incident #5 (n = 98 queries), with all models operating at inference time without weight updates.

5.1 Efficiency Gains

Increasing token reduction

ATLAS demonstrates systematic efficiency improvements as learned artifacts accumulate in the PLM and are retrieved in subsequent episodes (Figure 2). Overall, ATLAS averages 78,118 tokens per task, a 45% reduction relative to the autonomous GPT-5-mini baseline (141,660 tokens averaged over the 42 logged runs out of 47).

Phase breakdown reveals consistent learning progression:

- **Phase 1** (tasks 1–25): 100,810 tokens/task (–28.8% vs. baseline)
- **Phase 2** (tasks 26–60): 73,980 tokens/task (–47.8% vs. baseline)
- **Phase 3** (tasks 61–98): 67,002 tokens/task (–52.7% vs. basline)

Performance improvement

Efficiency gains coincide with stable mid-50% success: **52.0%** → **57.1%** → **52.6%**, demonstrating that ATLAS maintains accuracy while spending fewer tokens. The green trend line in Figure 2 reflects efficiency gains rising from 28.8% in Phase 1 to 52.7% in Phase 3, a +23.9 percentage point improvement. This demonstrates that reductions in token usage did not compromise accuracy, but instead facilitated more efficient and effective problem-solving with ATLAS.

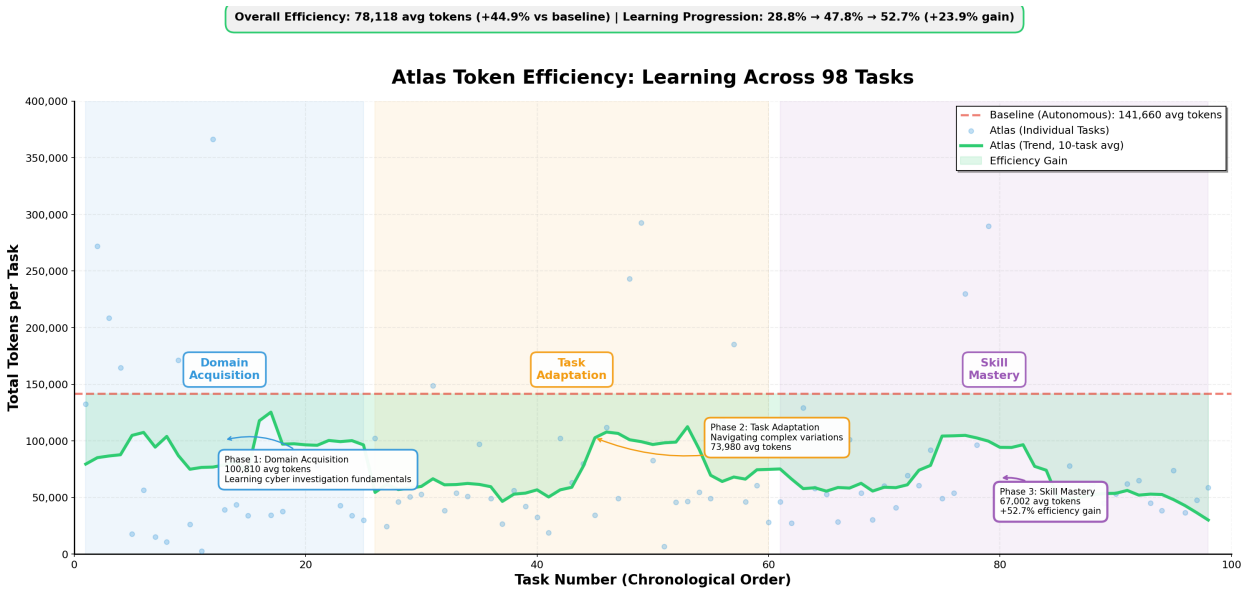

Figure 2: Continual learning progression across 98 tasks on ExCyTIn-Bench Incident #5. The learning journey unfolds in three phases: (1) Domain Acquisition (tasks 1-25) establishes foundational cyber investigation skills; (2) Task Adaptation (tasks 26-60) navigates diverse challenge variations; (3) Skill Mastery (tasks 61-98) achieves peak efficiency through consolidated learnings. Green trend line shows progressive improvement from 32.8% to 53.9% efficiency gain, demonstrating 21.1 percentage point improvement through inference-time continual learning.

5.2 Benchmark Performance

On Incident #5, ATLAS achieves **54.1% task success rate** (53/98 tasks) under the benchmark's scoring protocol. Compared to the GPT-5 (High) benchmark reference:

- **+6.1 percentage point higher success rate** (54.1% vs. 48.0% for GPT-5 High)
- **˜86% lower dollar cost per question** (≈$0.024 vs. $0.174 per question), using OpenAI's published GPT-5-mini and GPT-5 pricing tiers applied to each model's measured token usage.

These results demonstrate that ATLAS using a smaller model with inference-time learning, exceeds the performance of a larger baseline model at substantially lower computational cost (Wu et al., 2025).

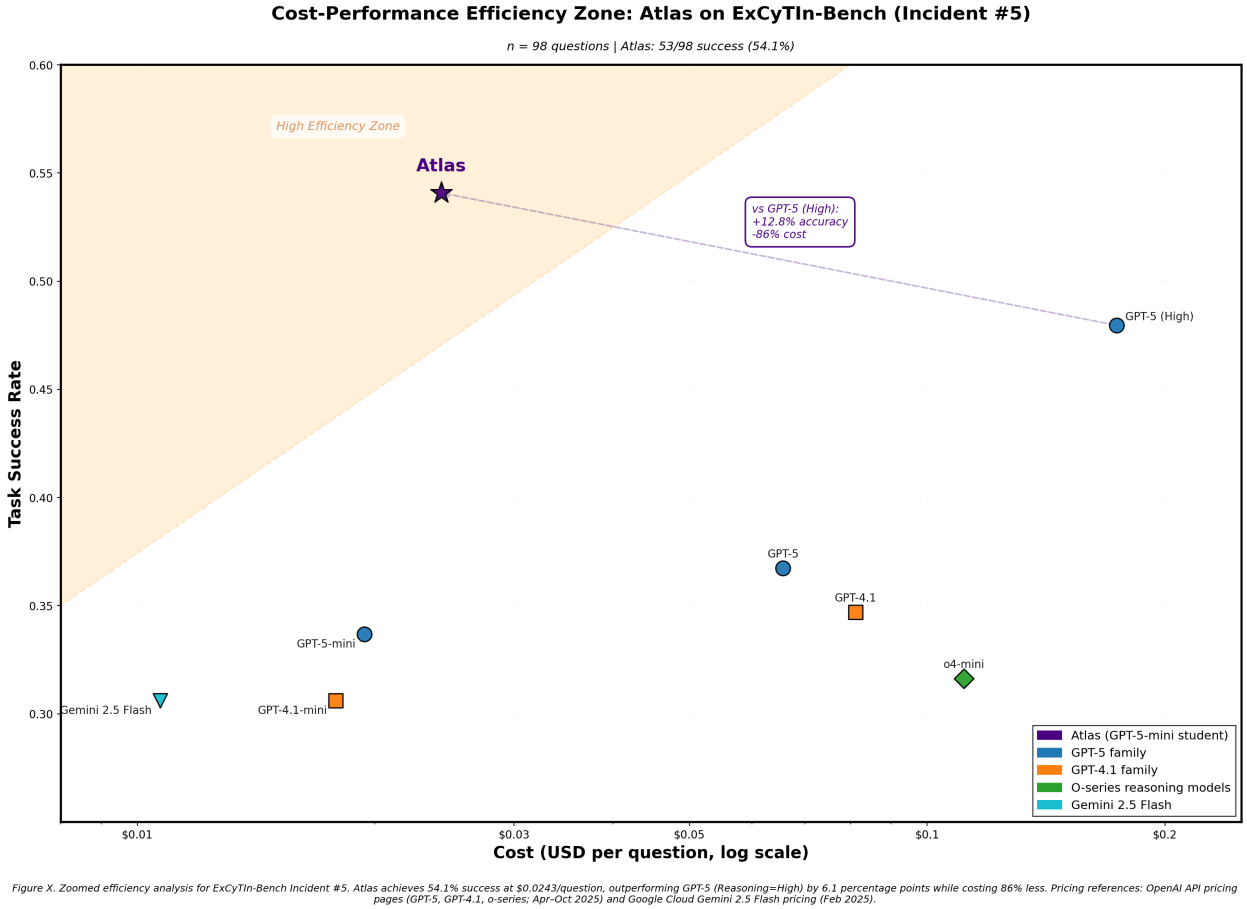

Figure 3: Performance and cost comparison on ExCyTIn-Bench (Incident #5). ATLAS (blue) achieves higher success (54.1%) at a lower cost (˜$0.024/task) than the stronger GPT-5 (High) baseline (48.0%, ˜$0.174/task).

Table 1: ExCyTIn-Bench (Incident #5) Performance Summary

Configuration	Success (Overall)	Avg Tokens / Task	Notes
ATLAS (Teacher & Student)	**54.1%**	**78,118**	GPT-5-mini student guided by GPT-5 teacher (n=98)
GPT-5 (Reasoning=High)	48.0%	71,105	Reproduced Incident #5 run (n=98)
GPT-5-mini (official baseline)	33.7%	61,562	Microsoft Incident #5 release (n=98)
GPT-5-mini (Student-only)	40.4%	141,660†	No pamphlets/teacher (n=47)

†Token averages computed over the 42/47 tasks with logged usage; success rate still counts all 47 tasks.

5.3 Cross-Incident Transfer (Incident 55)

To test whether guidance distilled on Incident #5 generalizes to new scenarios, we froze the learning memory and switched the runtime to auto mode (Student-only, no Teacher or Reward System). Running the same GPT-5-mini model on Incident #55, the official baseline clears 28 of 100 questions. With stored pamphlets injected into the Student's context, but no new Teacher feedback or reward signals, the model answers 41 correctly, a +46% improvement achieved purely through reused artifacts. This demonstrates that ATLAS learns transferable investigative strategies, not task-specific templates.

Output token analysis reveals efficient adaptation: completion tokens increase 50.3% (from 2,085 to 3,134 per question), but the composition shifts dramatically. The baseline produces 2,085 non-reasoning output tokens per question, while ATLAS generates only 999 non-reasoning tokens (−52.1%) and 2,135 reasoning tokens. Pamphlets guide the model toward deliberate reasoning rather than verbose exploration, reducing wasteful generation while increasing structured problem-solving. The output cost rises $0.002 per question, yielding 13 additional correct answers, or $0.016 per incremental success. While input prompt optimization remains necessary, the 28 → 41 accuracy gain provides our first empirical evidence that learning generalizes beyond the original incident domain.

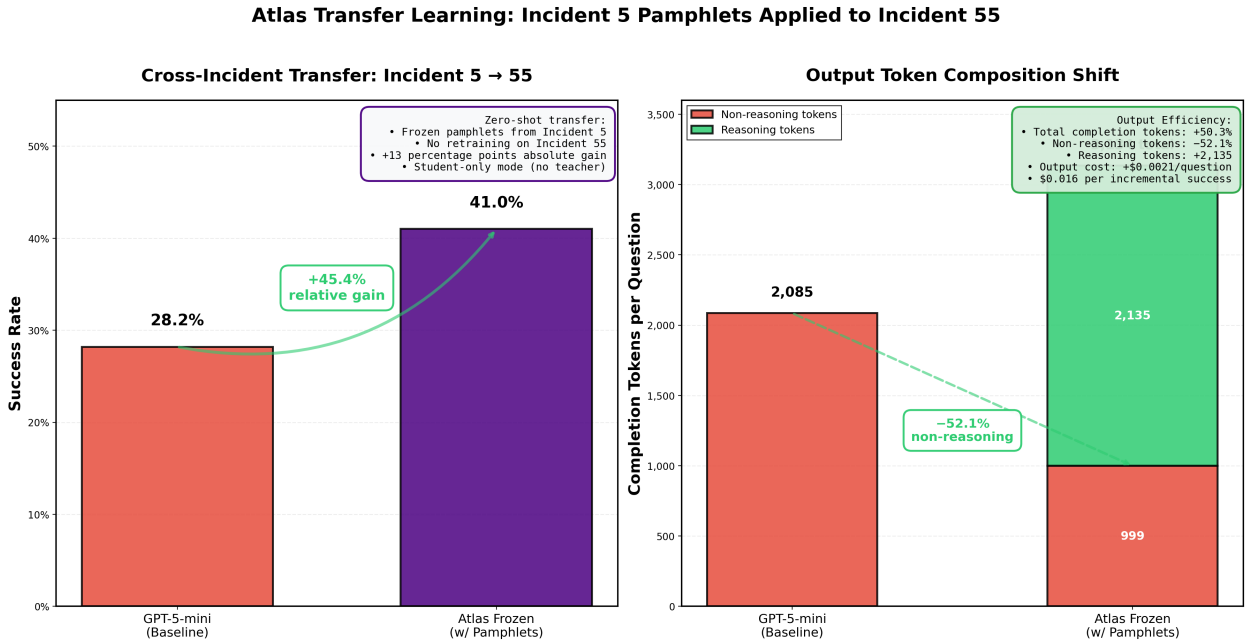

Figure X. Cross-incident transfer learning validation on ExCyTIn-Bench. Left: Atlas frozen mode (student-only with Incident 5 pamphlets) achieves 41% success on unseen Incident 55, improving 46% relatively over the 28.2% baseline GPT-5-mini agent (SecRL report, Table 2). Right: Output token composition shifts from verbose exploration to deliberate reasoning. Completion tokens increase 50.3% (2,085 → 3,134), but non-reasoning tokens decrease 52.1% (2,085 → 999) while reasoning tokens add 2,135. Atlas generates less wasteful output and more structured problem-solving. Output cost rises 0.0021perquestion, yielding0.016 per incremental success. This validates transferable investigative strategies with efficient adaptation.

Figure 4: Cross-incident transfer performance from Incident #5 to Incident #55. Using learned pamphlets from Incident #5, ATLAS improved success rate from 28% to 41% (+46%) with no additional training or real-time guidance.

6 Analysis

The observed efficiency and accuracy gains in ATLAS arise from two mechanisms: adaptive teaching and distilled experience transfer (DET).

Adaptive Teaching provides dynamic, context-aware guidance during task execution. By observing the Student's trajectory in real-time, the Teacher agent identifies and flags low-yield exploratory paths early, suggests high-value strategic pivots (e.g., schema-guided SQL filtering), and provides principle-level corrective guidance when the Student stalls. This adaptive supervision prunes the search space, preventing wasted computation on unproductive actions. The reward subsystem further reinforces effective strategies by associating Teacher interventions with principle-grounded rationales, helping to shape more efficient decision-making patterns over time. This mechanism mirrors interactive teaching, where support is adjusted based on the learner's immediate needs (Liu et al., 2023; Jung et al., 2025).

Distilled Experience Transfer (DET) enables cross-task learning and accelerates adaptation by leveraging codified knowledge from past interactions. Actionable guidance and successful strategies, distilled by a lightweight process

from Teacher interventions and high-reward trajectories, are stored as artifacts in the Persistent Learning Memory (PLM), indexed by task context. On subsequent, similar tasks, the orchestration layer surfaces this relevant distilled experience from the PLM. Applying this knowledge seeds the Student's initial plan or informs the runtime selection of the appropriate supervision level, allowing the system to bypass redundant exploration and rapidly apply proven tactics ("fast paths"). DET ensures that learnings from one episode are effectively transferred to future, related scenarios.

Together, **Adaptive Teaching** (real-time guidance) and **Distilled Experience Transfer** (leveraging past distilled lessons) create a synergistic effect. They enable the system to progressively shorten solution trajectories, reduce token consumption (Figure 2), and increase task success rates (Figures 5.2, 3).

Across the 98-task run, total tokens per task fall from 100,810 in the Domain Acquisition phase (tasks 1–25) to 67,002 in Skill Mastery (tasks 61–98) while success holds approximately 52–57%, demonstrating that ATLAS learns to solve cybersecurity incidents more economically without trading away accuracy.

Qualitative Example: Adaptation within a Single Incident

We trace the complete learning cycle through Incident #5, session 71 (dataset: `data/full_paired_trajectories.json`). The task required identifying the Security Identifier (SID) associated with suspicious remote activity on host `vnevado-win10r`.
Initial failure. The Student's first attempt produced an unverified answer and omitted the required structured reasoning trace. Critically, it failed to systematically inspect Windows security and incident telemetry tables, demonstrating a lack of principled investigation strategy.
Teacher intervention. Observing this failure, the Teacher issued principle-level guidance:

- Enumerate relevant telemetry sources before attempting attribution
- Prioritize tables: `DeviceProcessEvents`, `DeviceNetworkEvents`, `SecurityAlert`
- Join on host and trace identifiers; verify SID presence in returned records

The reward system evaluated this guidance with an auditable rationale and positive score. The learning engine then distilled the intervention into two complementary pamphlets:

- **Teacher pamphlet:** High-level investigative principles + diagnostic patterns
- **Student pamphlet:** Concrete SQL action schemas + validation guards

Successful re-execution. When the task was re-attempted with retrieved pamphlets seeding the context, the Student executed a systematic approach:

1. Issued `SHOW TABLES` and `DESCRIBE DeviceProcessEvents` to survey available schema
2. Filtered by target host: `DeviceName='vnevado-win10r'` with relevant temporal bounds
3. Constructed joins across `DeviceProcessEvents`, `DeviceNetworkEvents`, and `SecurityAlert`
4. Extracted and verified the correct SID: `S-1-5-21-1840191660-8534830288-125585561-1522`

The corrected trajectory satisfied the benchmark's success criterion while consuming fewer tokens than the initial attempt. A parallel autonomous execution of the same "SID of the account involved in the suspicious remote activity" prompt in `data/auto_trajectories.json` exhausted 304,389 tokens without ever landing on the canonical SID, underscoring that the retrieved pamphlet injected a reusable investigative pattern (schema scan $\rightarrow$ host filter $\rightarrow$ joined evidence check) rather than a cached answer. This principle-constrained search, guided by stored artifacts, is representative of the aggregate improvements observed across all 98 tasks in Incident #5.

6.1 Cross-Task Transfer Patterns

Trajectory analysis reveals systematic reuse of learned principles across heterogeneous tasks:

- **Guidance reuse spans disparate task types.** 69 of 98 trajectories include retrieved guidance in `metadata.secrl_applied_guidance`, and 68 of those inject skills (schema hygiene, constraint reconciliation, format discipline) that are absent from the new prompt text, showing that pamphlets capture abstract procedures rather than task-specific templates (`data/full_paired_trajectories.json[0..97].metadata`).
- **Overlap with autonomous runs shows true transfer.** On the 42 prompts that also appear in `data/auto_trajectories.json`, ATLAS succeeds 57.1% of the time while consuming 83.6k tokens on

average, versus 45.2% success at 144.6k tokens for the teacher-free baseline. Notably, 33 of those paired runs leveraged stored guidance, so higher accuracy derives from retrieved principles applied to new executions, not from repeated answers.

- **Process investigations become progressively cheaper.** The first three process-centric questions (indices 1, 6, 11) consume 217.7k tokens on average, whereas the last three (indices 90, 95, 96) finish in 48.8k tokens. Each later trajectory begins with the schema/prompting checklist minted during early failures, demonstrating that the same pamphlet shortens very different process-forensics tasks over time.

Together these measurements show that ATLAS stores reusable investigative strategies in its PLM and redeploys them across heterogeneous prompts, which in turn explains the steady efficiency gains in Figure 2.

6.2 Reproducibility

To enable independent verification and extension of our results, we have released a comprehensive supplementary dataset. This dataset includes complete session traces documenting timestamped action sequences across all 98 tasks, alongside full Teacher interventions that capture the rationales and guidance provided at each decision point. We provide reward annotations containing principle-grounded justifications paired with numerical scores, as well as all generated Learning Pamphlets with their associated metadata. Finally, we include detailed token accounting logs that align precisely with the benchmark's official scoring configuration, permitting complete audit of our reported metrics and decision paths.

7 Future Research

Our work on system-centric CL opens several research directions that can be explored, particularly given its unique advantages in accessibility, deployability, and practical real-world applicability. We highlight four key areas in the following subsections.

7.1 Architectural Design Exploration

We will continue to compare and study alternative system-based designs remains essential. Comparative studies examining multi-agent ensembles, hierarchical memory structures, and varied capability probing mechanisms would elucidate trade-offs between learning speed, computational overhead, and architectural complexity. A particularly compelling question is: Can corrective strategies learned by one system be transferred or hierarchically combined across different architectures?

7.2 Knowledge Generalization

The principles stored in ATLAS's persistent memory suggest opportunities for cross-model and cross-task generalization. Can Teacher-generated corrective feedback trained on one Student model accelerate adaptation when transferred to other agents? Developing principled methods for distilling, validating, and transferring learned strategies could dramatically enhance the portability and scalability of system-centric learning, potentially creating reusable "libraries" of adaptive principles.

7.3 Adaptive Evaluation Methodologies

As we build systems that learn continuously, static benchmarks become insufficient. Our future work will focus on creating dynamic benchmarks that adapt alongside the agent, potentially increasing difficulty or introducing novel scenarios based on agent performance, and developing robust metrics to measure adaptation beyond simple task success, such as resilience to distribution shifts or efficiency of knowledge acquisition as mitigating evaluation hacking remains a key challenge.

7.4 Hybrid Online and Offline Learning

Another promising research direction lies in integrating world models trained offline on ATLAS-generated execution traces back into the live system. These learned models could serve as predictive simulators for counterfactual planning, as structured knowledge sources augmenting Teacher reasoning, or as components enabling more sophisticated credit assignment. Such world models could serve as planning simulators, Teacher enhancement modules, or sources of structured causal priors. By closing this loop between online and offline learning, ATLAS could evolve into a hybrid

continual learner, one that couples immediate, inference-time adaptation with deeper, model-based understanding (Yu et al., 2023).

8 Conclusion

We challenge the dominant model-centric paradigm in Continual Learning (CL), arguing that it fails to meet the demands of dynamic, real-world deployment. In such environments, adaptation cannot depend on offline retraining. True adaptability must occur at inference time.

We proposed system-centric CL with the defined goal of achieving efficient performance, measurable improvements in task success together with reductions in computational cost at inference-time.

We introduced ATLAS, a dual-agent architecture that operationalizes this paradigm. By orchestrating Student and Teacher models around a persistent learning memory, ATLAS achieves gradient-free adaptation, bypassing the need for retraining, specialized hardware, or adaptation-specific datasets. This democratizes continual learning, making it accessible for deployment on standard inference infrastructure.

Our evaluation on ExCyTIn-Bench validates this approach. ATLAS improved task success from 33.7% (benchmark GPT-5-mini) / 40.4% (Student-only) to 54.1% as experience accumulated, while reducing the Student's average tokens from 141,660 to 78,118 (−45%). The adapted system using GPT-5-mini as Student achieved 54.1% task completion, surpassing the 48.0% baseline of the larger GPT-5 (High) model while operating at the GPT-5-mini cost tier. These results demonstrate that system-centric learning matches or exceeds compute-intensive training methods under strict inference-time constraints.

Beyond performance, our work highlights the urgent need for adaptive evaluation methods capable of assessing dynamically learning systems, a critical gap as the field moves beyond static models. Furthermore, we demonstrated that the causally-annotated traces generated by ATLAS's learning process provide a powerful data engine for training world models, bridging online adaptation and offline model building.

In summary, system-centric CL offers a accessible, scalable, and immediately deployable path toward AI systems that improve through use learning efficiently and incrementally within the environments they are deployed.

References

Agrawal, L. A., Tan, S., Soylu, D., Ziems, N., Khare, R., Opsahl, T., Khodabakhsh, A., Wang, S., Tadaka, E., Wallace, E., Li, M., Sabharwal, A., Potts, C., & Khattab, O. (2025). GEPA: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*.

Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*.

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., van den Driessche, G., Lespiau, J.-B., Damoc, B., Clark, A., De Las Casas, D., Guy, A., Menick, J., Ring, R., Hennigan,T., Huang, S., Maggiore, L., Jones, C., Cassirer, A., Brock, A., Paganini, M., Irving, G., Vinyals, O., Osindero, S., Simonyan, K., Rae, J. W., Elsen, E., & Sifre, L. (2022). Improving language models by retrieving from trillions of tokens (RETRO). *Proceedings of Machine Learning Research, 162*, 2206–2240.

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. *arXiv preprint arXiv:2305.14314*.

Frantar, E., & Alistarh, D. (2023). SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning* (PMLR, Vol. 202, pp. 10323–10337).

Guo, D., Rush, A. M., & Kim, Y. (2021). Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (pp. 4884–4896). Association for Computational Linguistics.

Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M.-W. (2020). REALM: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning* (PMLR, Vol. 119, pp. 3929–3938).

Ha, D., & Schmidhuber, J. (2018). World models. *arXiv preprint arXiv:1803.10122*.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jung, J., Brahman, F., & Choi, Y. (2025). Trust or escalate: LLM judges with provable guarantees for human agreement. In *International Conference on Learning Representations (ICLR)*.

Khattab, O., Singhvi, A., Maheshwari, P., Zhang, Z., Santhanam, K., Vardhamanan, S., Haq, S., Sharma, A., Joshi, T. T., Moazam, H., Miller, H., Zaharia, M., & Potts, C. (2Next). DSPy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences, 114*(13), 3521–3526.

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 3045–3059). Association for Computational Linguistics.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459–9474.

Lin, X. V., Chen, X., Chen, M., Shi, W., Lomeli, M., James, R., Rodriguez, P., Kahn, J., Szilvasy, G., Lewis, M., Zettlemoyer, L., & Yih, W.-T. (2Next). RA-DIT: Retrieval-augmented dual instruction tuning. In *International Conference on Learning Representations (ICLR)*.

Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., & Zhu, C. (2023). G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Liu, S.-Y., Wang, C.-Y., Yin, H., Wang, Y.-C. F., Cheng, K.-T., & Chen, M.-H. (2024). DoRA: Weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning* (PMLR, Vol. 235).

Malik, S., Pyatkin, V., Land, S., Morrison, J., Smith, N. A., Hajishirzi, H., & Lambert, N. (2025). RewardBench 2: Advancing reward model evaluation. *arXiv:2506.01937*.

Packer, C., Wooders, S., Lin, K., Fang, V., Patil, S. G., Stoica, I., & Gonzalez, J. E. (2023). MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*.

Pham, Q., Liu, C., & Hoi, S. C. H. (2021). DualNet: Continual learning, fast and slow. In *Advances in Neural Information Processing Systems, 34*.

Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., & Wayne, G. (2019). Experience replay for continual learning. *Advances in Neural Information Processing Systems, 32*, 350–360. `https://papers.nips.cc/paper/2019/hash/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Abstract.html`

Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems, 36*.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., & Anandkumar, A. (2023). Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Wu, Y., Velazco, M., Zhao, A., Meléndez Luján, M. R., Movva, S., Roy, Y. K., Nguyen, Q., Rodriguez, R., Wu, Q., Albada, M., Kiseleva, J., & Mudgerikar, A. (2025). ExCyTIn-Bench: Evaluating LLM agents on cyber threat investigation. *arXiv preprint arXiv:2507.14201*.

Yu, T., Ruan, S., & Xing, E. P. (2023). Explainable reinforcement learning via a causal world model. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*. `https://arxiv.org/abs/2305.02749`

Zhou, A., Yan, K., Shlapentokh-Rothman, M., Wang, H., & Wang, Y.-X. (2024). Language agent tree search unifies reasoning, acting, and planning in language models (LATS). *arXiv:2310.04406*.