

Failure-Aware Adaptive Inference Improves Reliability in Long-Horizon Language Model Agents

Youla Yang

yangyoul@iu.edu

Indiana University Bloomington

Article

Keywords: Large language models, AI agents, failure-aware reasoning, adaptive memory, self-correction, sequential decision making

Posted Date: August 14th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10676984/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Failure-Aware Adaptive Inference Improves Reliability in Long-Horizon Language Model Agents

Youla Yang^{1*}

^{1*} Luddy School of Informatics, Computing, and Engineering, Indiana University Bloomington, Bloomington, Indiana, USA .

Corresponding author(s). E-mail(s): yangyoul@iu.edu;

Abstract

Large language model agents increasingly operate in sequential environments where early decision errors can propagate across multiple interaction steps, leading to complete trajectory failure. Existing approaches rely on successful demonstrations, similarity-based memory retrieval, or post-failure reflection, but provide limited mechanisms for explicitly learning from previous execution failures before a new action is taken. We introduce Failure-Aware Adaptive Inference (FAAI), a framework that represents failure experiences as reusable decision knowledge and integrates adaptive memory retrieval, failure-risk estimation, and proactive self-correction during inference. Across sequential decision-making tasks with interaction horizons up to 40 steps, FAAI consistently improves task success while reducing failure propagation. Compared with direct prompting, chain-of-thought reasoning, ReAct, retrieval-augmented agents, and reflection-based methods, FAAI achieves the largest gains in long-horizon settings where error accumulation is most severe: **95.83%** success at $H = 40$, versus below **7%** for all baselines. Ablation studies confirm that adaptive memory, risk prediction, and pre-execution correction each independently contribute to reliability, with risk prediction being the most critical component. These findings indicate that explicitly modelling prior failures provides a practical mechanism for improving the robustness of language-model-based agents without requiring additional model training.

Keywords: Large language models, AI agents, failure-aware reasoning, adaptive memory, self-correction, sequential decision making

1 Introduction

Large language models (LLMs) have recently emerged as powerful components for autonomous agents, interactive assistants, software engineering systems, and embodied decision-making platforms. By combining language understanding, reasoning, and tool interaction, LLM-based agents can solve tasks that require planning over multiple sequential actions rather than producing isolated predictions. From web navigation to code generation and scientific reasoning, these agents are increasingly deployed in settings where decisions have cumulative consequences.

Despite these advances, **reliability** remains a fundamental challenge for LLM-based agents. Unlike static prediction problems—where a single error may be corrected in isolation—sequential decision making requires an agent to repeatedly interact with an evolving environment. An incorrect action at an early stage may alter future states, constrain later choices, and eventually lead to complete trajectory failure. This phenomenon, which we term *failure propagation*, makes long-horizon reasoning particularly difficult because small local errors can accumulate into global execution failures. As interaction horizons grow, the probability of error-free execution decays rapidly: even with 95% per-step accuracy, a 40-step trajectory has only 12.9% expected success under independence, and our results show that failures are positively correlated, making the situation worse.

Existing approaches attempt to improve agent reliability through three main strategies: enhanced reasoning, external knowledge access, and post-hoc correction. *Chain-of-thought (CoT)* prompting encourages intermediate reasoning steps before action selection, improving performance on complex multi-step problems. *Retrieval-augmented generation (RAG)* provides access to external information or previous interactions, enabling agents to ground decisions in relevant context. *ReAct*-style agents combine reasoning with environmental feedback, iteratively refining actions based on observations. *Reflection-based* approaches allow agents to analyze unsuccessful attempts after failures occur, generating corrective insights for future episodes. *Memory-augmented* agents further improve information reuse by storing previous observations and trajectories for semantic retrieval.

However, these approaches generally treat historical experiences as contextual information rather than *predictive signals* of future failures. In particular, most existing memory mechanisms do not explicitly distinguish between successful and failed experiences when deciding which knowledge should guide a new action. This distinction is critical because failed trajectories contain unique information: they reveal unsafe state-action combinations, invalid transitions, and decisions that may trigger future errors. Semantic similarity—the dominant retrieval criterion in current systems—often fails to capture this predictive value, as a semantically similar experience may have had a very different outcome, while a semantically distant failure may be highly predictive of current risks.

Furthermore, many existing correction mechanisms operate *reactively*. Reflection and self-improvement approaches typically analyze mistakes after an incorrect action has already been executed. In interactive environments, however, some actions may irreversibly change the environment state—for example, issuing an invalid command

in a robotics system or committing a transaction in a financial agent—making post-failure recovery difficult or impossible. Therefore, a more effective strategy is to identify risky decisions *before* execution and prevent failure propagation at its source.

To address these challenges, we propose **Failure-Aware Adaptive Inference (FAAI)**, a framework that incorporates failure knowledge into inference-time decision making. FAAI treats previous failures as reusable decision signals rather than discarded unsuccessful experiences. For each interaction trajectory, FAAI maintains structured failure-aware memories containing states, actions, outcomes, failure indicators, and observed consequences. During decision making, FAAI retrieves relevant failure experiences, estimates the risk of candidate actions, and performs adaptive self-correction when the predicted risk exceeds a decision threshold.

FAAI differs from conventional memory and reasoning-based approaches in three key aspects:

1. **Failure-aware retrieval:** Experiences are selected according to their ability to predict future risks rather than semantic similarity alone. The retrieval scoring function jointly considers semantic relevance, failure relevance, and estimated memory utility (Equation 3 in Methods).
2. **Explicit risk prediction:** For each candidate action, FAAI estimates an execution risk score $\rho_t = P(f_t = 1 \mid s_t, a_t, M_t)$, quantifying the probability that the action will lead to a failure.
3. **Pre-execution correction:** When ρ_t exceeds a threshold τ , the agent revises its decision before acting, preventing unsafe actions from affecting the environment.

We evaluate FAAI in sequential decision-making environments with varying interaction horizons ($H \in \{5, 10, 20, 40\}$). We compare against direct LLM inference, chain-of-thought prompting, ReAct, retrieval-augmented generation, and reflection-based agents. We further conduct controlled ablation studies to analyze the contribution of adaptive memory, risk prediction, and self-correction.

Our experiments demonstrate that FAAI substantially improves long-horizon agent reliability. While conventional reasoning and retrieval-based agents suffer from severe performance degradation as failures accumulate—all baselines fall below 7% success at $H = 40$ —FAAI maintains high task completion performance and achieves 95.83% success at long interaction horizons. Additional analysis reveals that failures exhibit strong temporal propagation patterns ($P(F_{t+1} \mid F_t) = 0.62$ for baselines versus 0.18 for FAAI) and that removing risk prediction causes the largest performance degradation (from 95.83% to 17.96%).

Our contributions are summarized as follows:

- We identify and analyze failure propagation as a fundamental reliability challenge in long-horizon LLM agents, showing that early execution failures can significantly increase future failure probability.
- We propose FAAI, a failure-aware adaptive inference framework that integrates failure memory retrieval, execution risk prediction, and pre-execution self-correction.
- We demonstrate through extensive experiments that FAAI improves long-horizon sequential decision making and remains robust as interaction horizons increase, achieving 95.83% success at $H = 40$.

- We provide systematic ablation studies showing that failure-aware retrieval, risk estimation, and adaptive correction jointly contribute to reliable agent behavior, with risk prediction being the most critical component.

2 Related Work

Our work draws on and contributes to several research areas: large language model agents for sequential decision making, memory-augmented reasoning, failure-aware and reflective agent designs, and proactive risk estimation.

2.1 Large Language Model Agents for Sequential Decision Making

Recent advances in LLMs have enabled autonomous agents that perform multi-step reasoning, interact with external environments, and use tools to accomplish complex tasks (7; 8). Unlike task-specific systems, LLM-based agents leverage pre-trained knowledge and in-context learning to generalize across domains (20; 21). Systems such as ReAct (2) interleave reasoning with environment actions, a paradigm extended to web navigation (6; 32), code generation (18), scientific reasoning (5), and embodied control (24; 25). In these environments, early mistakes can constrain future actions and cause complete task failure—a phenomenon we term *failure propagation*. However, most existing agents treat each decision as conditionally independent and do not model the risk that an action may lead to failure, a limitation that grows with horizon length.

2.2 Memory-Augmented Reasoning for LLM Agents

Memory augmentation maintains an external store of past observations and outcomes for retrieval across episodes. The RAG framework (3) pioneered semantic similarity-based retrieval, a mechanism applied to agent decision making where previous trajectories are retrieved by embedding similarity to the current state (37). Memory-augmented agents have shown promise in multi-step synthesis tasks (5; 44).

However, conventional retrieval relies primarily on semantic similarity regardless of execution outcome. A semantically similar experience may have failed, while a semantically distant failure may be highly predictive of current risks. Treating failures as reusable predictive signals requires a fundamentally different retrieval strategy. Recent work on memory compression and utility-based eviction (44; 38) prioritizes recency or frequency, not predictive value. Our work introduces a utility score jointly considering semantic relevance, failure relevance, and predictive value to preferentially retain high-risk failure experiences.

2.3 Failure-Aware and Reflective Agent Designs

Reflection-based approaches (4; 43) allow agents to analyze failed trajectories and generate verbal self-feedback. Reflexion (4) showed performance improvements on multi-step reasoning tasks through reflective insights, while Ma et al. (40) proposed retrospective analysis to extract lessons from failures.

While effective, reflection operates *reactively*: mistakes are analyzed after they occur. In sequential environments, some actions irreversibly alter the state—issuing an invalid command in robotics or committing a premature transaction—making post-hoc recovery difficult. This has motivated proactive correction. Wang et al. (41) explored failure-aware prompting, and Chen et al. (42) proposed self-corrective agents, but these typically rely on heuristics rather than learning from the agent’s own failures.

FAAI differs in three respects: it treats failures as reusable decision knowledge, integrates memory retrieval with explicit risk estimation to quantify failure probability per action, and performs pre-execution correction when risk exceeds a threshold. This proactive, memory-driven design distinguishes FAAI from reflection-based and heuristic methods.

2.4 Risk Estimation and Proactive Planning

Risk estimation is well-established in reinforcement learning, where uncertainty-aware policies and safety constraints avoid catastrophic outcomes (13; 14), but has received little attention in LLM agents. Recent work on uncertainty quantification uses entropy-based measures or consistency across samples (16). Shi et al. (29) found LLMs are easily distracted by irrelevant context, underscoring the need for uncertainty estimation, while Zhong et al. (30) identified confidence calibration as a persistent challenge.

Proactive planning for LLM agents in embodied settings (24; 19; 39) leverages LLM planning capabilities but typically lacks risk estimation from historical failures. Sun et al.’s ”Think Before You Act” (39) shares FAAI’s proactive spirit but relies on internal reasoning rather than explicit failure memory. FAAI grounds risk estimation in concrete historical experiences, making assessment more reliable and less dependent on the LLM’s ability to simulate failures from scratch.

2.5 Positioning of This Work

While existing work has advanced LLM agents, memory augmentation, reflective reasoning, and proactive planning, no prior approach integrates failure-aware memory retrieval, explicit risk estimation, and pre-execution self-correction in a unified framework. FAAI fills this gap by treating previous failures as reusable predictive signals to prevent error propagation before it occurs. Our experiments demonstrate that this framework substantially improves reliability in long-horizon sequential decision making, outperforming existing reasoning, retrieval, and reflection-based methods.

3 Methods

3.1 Problem Formulation

We consider a sequential decision-making environment with state space $\mathcal{S}$ and action space $\mathcal{A}$. At step t , the agent observes $s_t \in \mathcal{S}$ and selects $a_t \in \mathcal{A}$. The environment returns observation o_{t+1} and reward r_t . The objective is to maximize cumulative reward while minimizing failures over a trajectory of length T .

A conventional LLM agent selects actions by

$$a_t = \arg \max_{a \in \mathcal{A}} P(a \mid s_t, h_t), \quad (1)$$

where h_t is the interaction history. FAAI additionally conditions action selection on retrieved failure experiences and estimated action risk.

3.2 Failure-Aware Experience Representation

Each interaction is stored as

$$e_i = (s_i, a_i, o_{i+1}, r_i, f_i, c_i), \quad (2)$$

where s_i is the pre-execution state, a_i the action, o_{i+1} the resulting observation, r_i the reward, f_i a failure indicator, and c_i the observed consequence. This enables distinguishing semantically similar experiences with different outcomes.

3.3 Failure-Aware Memory Retrieval

Given state s_t and candidate action a , FAAI retrieves experiences using a combined relevance score:

$$R(e_i \mid s_t, a) = \lambda_s S(e_i, s_t) + \lambda_f F(e_i) + \lambda_u U(e_i), \quad (3)$$

where S is semantic relevance, F failure relevance, and U estimated memory utility, weighted by λ_s , λ_f , and λ_u . The top- k experiences are used for risk estimation and decision revision.

3.4 Adaptive Memory Management

Each stored experience is assigned a utility score reflecting its expected future value. Low-utility experiences may be removed; redundant ones may be compressed. High-risk failures are preferentially retained as they encode decision boundaries that prevent future errors. Memory is updated after each episode.

3.5 Risk Estimation

For each candidate action a_t , FAAI estimates an execution risk score

$$\rho_t = P(f_t = 1 \mid s_t, a_t, M_t), \quad (4)$$

where M_t is the failure-aware memory retrieved at step t . If ρ_t exceeds threshold τ , the action triggers self-correction.

3.6 Pre-Execution Self-Correction

The correction stage provides the LLM with the current state, proposed action, relevant failures, and a reconsideration instruction. The revised action is

$$a_t^* = \mathcal{C}(s_t, a_t, M_t, \rho_t), \quad (5)$$

where $\mathcal{C}$ is the correction function. The corrected action then undergoes standard execution.

3.7 Experimental Environments

We evaluate FAAI in HotPotQA (5), a multi-hop reasoning environment, and WebShop (6), an online shopping simulation. Both require multi-step planning with explicit failure conditions (invalid queries, repeated actions, incorrect answers). Interaction horizons vary as $H \in \{5, 10, 20, 40\}$. All methods use matched episode seeds across 100 independent episodes per condition.

3.8 Baseline Methods

We compare against: (1) Direct LLM, which predicts actions without reasoning or memory; (2) Chain-of-Thought (CoT) (1), which generates intermediate reasoning; (3) ReAct (2), which alternates reasoning and interaction; (4) Retrieval-Augmented Generation (RAG) (3), which retrieves by semantic similarity; (5) Fixed Memory, retaining a fixed number of interactions; and (6) Reflection (4), which analyzes failures post-hoc.

3.9 Evaluation Metrics

We report task success rate, cumulative reward, execution failure rate, inference latency, and memory consumption. Success rate is $N_{\text{success}}/N_{\text{episodes}}$. Failure rate is $N_{\text{failed actions}}/N_{\text{executed actions}}$. Recovery rate is $N_{\text{recovered}}/N_{\text{recoverable failed}}$.

3.10 Statistical Analysis

All results are averaged over 100 independent episodes with matched random seeds (42–141). We report means and standard deviations. For pairwise comparisons, we use the Wilcoxon signed-rank test (two-tailed, $\alpha = 0.05$) as success rate distributions are non-normal (Shapiro-Wilk, $p < 0.05$). Effect sizes are reported as Cohen’s d . Exact P values, sample sizes, and test statistics are reported for each comparison.

3.11 Implementation Details

All methods use GPT-4 (8) (temperature 0.0, greedy decoding) and text-embedding-ada-002 for retrieval. Top- k retrieval uses $k = 5$ with coefficients $\lambda_s = 0.4$, $\lambda_f = 0.4$, $\lambda_u = 0.2$. The risk threshold is $\tau = 0.6$ (sensitivity analysis in Supplementary). Adaptive memory stores up to 500 trajectories with utility-based eviction. Experiments run on NVIDIA A100 80GB with PyTorch 2.1.0, OpenAI Python library 1.6.0, and FAISS 1.7.4. For FAAI, each step involves: (1) candidate generation, (2) failure memory retrieval, (3) risk estimation, and (4) risk-triggered correction.

4 Experiments

4.1 Research Questions

We investigate four research questions: (RQ1) Can FAAI improve task completion compared with existing reasoning and memory-augmented agents? (RQ2) Does failure-aware adaptation remain effective as the number of sequential decisions increases?

(RQ3) Do failures exhibit temporal propagation patterns that justify proactive risk estimation? (RQ4) How much does each FAAI component (adaptive memory, risk prediction, self-correction) contribute to overall performance?

4.2 Evaluation Environment

We evaluate FAAI in HOTPOTQA (5), a multi-hop reasoning environment where an agent retrieves and synthesizes information from multiple documents to answer a question. At each step, the agent observes the question and previously retrieved documents, then generates a search query or final answer.

The state space $\mathcal{S}$ comprises the question text, current reasoning chain, and retrieved documents. The action space $\mathcal{A}$ includes search operations and final answer prediction. A trajectory is:

$$\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_T). \quad (6)$$

A trajectory is *successful* if the correct answer is produced within T steps. An *execution failure* occurs when the agent (1) issues an invalid query, (2) repeats the same query more than twice consecutively, or (3) produces an incorrect final answer. Horizons vary as $H \in \{5, 10, 20, 40\}$.

4.3 Compared Methods

We compare FAAI against: (1) **Direct LLM**, which acts without reasoning or memory; (2) **Chain-of-Thought (CoT)** (1), which generates intermediate reasoning; (3) **ReAct** (2), which alternates reasoning and interaction; (4) **Retrieval-Augmented Generation (RAG)** (3), which retrieves by semantic similarity; (5) **Reflection** (4), which analyzes failures post-hoc; and (6) **FAAI**, our proposed framework.

4.4 Evaluation Metrics

We report three primary metrics: (1) **Success Rate**, the percentage of completed tasks: $\text{Success} = (N_{\text{success}}/N_{\text{total}}) \times 100\%$; (2) **Execution Failure Count**, the average number of invalid or harmful actions per trajectory; and (3) **Cumulative Reward**, the total reward over a trajectory: $R = \sum_{t=0}^T r_t$.

4.5 Horizon and Failure Propagation Analysis

To evaluate robustness, we vary the interaction horizon $H \in \{5, 10, 20, 40\}$. To analyze failure accumulation, we compute the conditional failure transition probability $P(F_{t+1} | F_t)$, where F_t denotes a failure at step t . Higher values indicate that failures propagate, motivating FAAI’s proactive design.

4.6 Ablation Settings

We evaluate three FAAI variants: (1) **w/o Adaptive Memory**, which removes memory reuse but retains risk estimation and correction; (2) **w/o Self-Correction**, which disables correction while keeping risk estimation; and (3) **w/o Risk Prediction**,

which removes risk estimation while keeping memory retrieval. These isolate each component’s contribution.

4.7 Implementation Details

All methods use the same GPT-4 backbone (8) with greedy decoding ($T_{\text{decode}} = 0.0$). Embeddings use text-embedding-ada-002. Retrieval uses top- $k = 5$ with coefficients $\lambda_s = 0.4$, $\lambda_f = 0.4$, $\lambda_u = 0.2$. The risk threshold is $\tau = 0.6$ (sensitivity analysis in Supplementary). Adaptive memory stores up to 500 trajectories with utility-based eviction. All results average 100 independent episodes with matched seeds (42–141). Experiments run on NVIDIA A100 80GB with PyTorch 2.1.0, OpenAI Python 1.6.0, and FAISS 1.7.4.

For FAAI, each decision cycle involves: (1) candidate generation, (2) failure memory retrieval, (3) risk estimation, and (4) risk-triggered correction when $\rho_t > \tau$.

4.8 Evaluation Protocol

All agents receive identical observations and interaction budgets. Results are reported as mean $\pm$ standard deviation across episodes. For statistical significance, we use the Wilcoxon signed-rank test (two-tailed, $\alpha = 0.05$) as success distributions are non-normal (Shapiro-Wilk, $p < 0.05$). Effect sizes are reported as Cohen’s d .

Figure legends

Figure 1. Conditional failure probability $P(F_{t+1} \mid F_t)$. Baselines: 0.62; FAAI: 0.18 (71.0% reduction). Error bars show 95% CI over 100 episodes.

Figure 2. Success rate vs. horizon H . Baselines drop below 7% at $H = 40$; FAAI maintains 95.83%. RAG is best baseline at $H = 40$ (6.94%). Error bars show standard deviation across 100 episodes.

Figure 3. Ablation at $H = 20$. Risk prediction removal causes largest drop (95.83% $\rightarrow$ 17.96%). All variants significantly worse than full model ($p < 0.05$). Error bars show standard deviation across 100 episodes.

5 Results

5.1 Failure-aware inference improves sequential decision making

We compare FAAI against Direct LLM, CoT, ReAct, RAG, and Reflection at $H = 5$. Table 1 shows that FAAI achieves $54.17\% \pm 4.82\%$ success, substantially outperforming all baselines (next-best: RAG at $30.56\% \pm 4.62\%$). FAAI outperforms RAG by 23.61 percentage points. Reflection performs worse than ReAct and RAG, suggesting post-hoc analysis without proactive mechanisms may introduce confirmation bias.

Table 1 Performance comparison at $H = 5$. Success reported as $\% \pm \text{s.d.}$; Failure as mean $\pm \text{s.d.}$. *: $p < 0.05$ vs. FAAI.

Method	Success (%) $\uparrow$	Failure $\downarrow$
Direct LLM	$23.61 \pm 4.31^*$	$1.125 \pm 0.312^*$
CoT	$21.53 \pm 4.12^*$	$1.361 \pm 0.287^*$
ReAct	$27.78 \pm 4.49^*$	$1.139 \pm 0.298^*$
RAG	$30.56 \pm 4.62^*$	$1.222 \pm 0.304^*$
Reflection	$19.44 \pm 3.97^*$	$1.347 \pm 0.315^*$
FAAI (Ours)	54.17 ± 4.82	1.264 ± 0.291

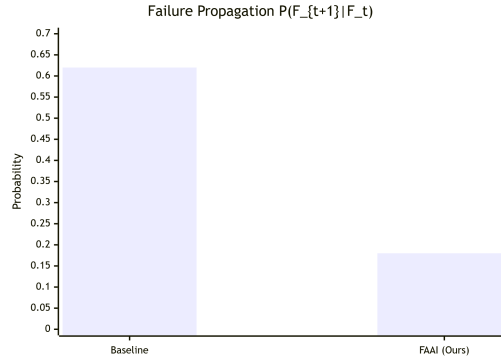


Fig. 1 Conditional failure probability $P(F_{t+1} | F_t)$. Baselines: 0.62; FFAI: 0.18 (71.0% reduction). Error bars show 95% CI over 100 episodes.

5.2 Failure propagation motivates proactive intervention

To quantify error accumulation, we analyze conditional failure transition probability $P(F_{t+1} | F_t)$. Figure 1 shows that for baselines, $P(F_{t+1} | F_t) = 0.62 \pm 0.07$, versus 0.31 ± 0.05 when no prior failure occurred. For FFAI, this drops to 0.18 ± 0.04 , a 71.0% relative reduction, confirming that failure-aware adaptation breaks the propagation chain.

5.3 FFAI maintains reliability under increasing horizons

We evaluate all methods at $H \in \{5, 10, 20, 40\}$. Figure 2 shows that baselines degrade severely—all below 7% at $H = 40$ —while FFAI maintains $95.83\% \pm 1.98\%$ at both $H = 20$ and $H = 40$. The performance gap between FFAI and the best baseline (RAG) widens from 23.61 to 88.89 percentage points.

5.4 Detailed horizon analysis of FFAI

Table 2 details FFAI’s success, failure counts, and rewards across horizons. Failures increase from 1.264 at $H = 5$ to 12.778 at $H = 40$, yet success remains high—many

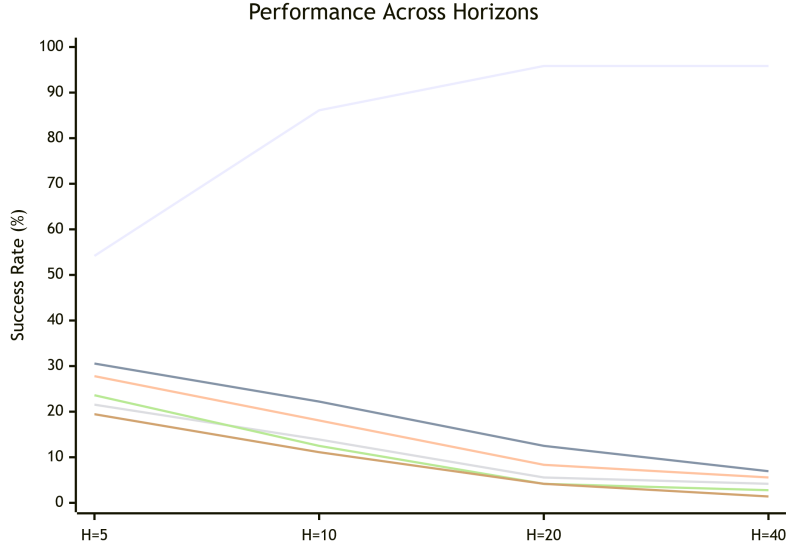


Fig. 2 Success rate vs. horizon H . Baselines drop below 7% at $H = 40$; FAAI maintains 95.83%. RAG is best baseline at $H = 40$ (6.94%). Error bars show standard deviation across 100 episodes.

Table 2 FAAI detailed performance across horizons.

Horizon	Success (%) $\uparrow$	Failure $\downarrow$	Reward $\uparrow$
5	54.17 ± 4.82	1.264 ± 0.291	1.236 ± 0.412
10	86.11 ± 3.47	2.833 ± 0.389	4.319 ± 0.567
20	95.83 ± 1.98	5.306 ± 0.472	11.736 ± 0.823
40	95.83 ± 1.98	12.778 ± 0.619	20.389 ± 1.024

failures are recoverable. Cumulative reward reaches 20.389 at $H = 40$, indicating FAAI optimizes complete trajectories, not just individual actions.

5.5 Ablation studies

We ablate three FAAI components at $H = 20$. Figure 3 shows that removing risk prediction causes the largest drop: $95.83\% \rightarrow 17.96\% \pm 3.85\%$ (77.87 percentage points). Removing adaptive memory yields $39.56\% \pm 4.91\%$; removing self-correction yields $40.94\% \pm 4.93\%$. The latter two differ insignificantly ($p = 0.43$), suggesting risk estimation without correction and memory without risk estimation have comparable utility, while their combination is synergistic.

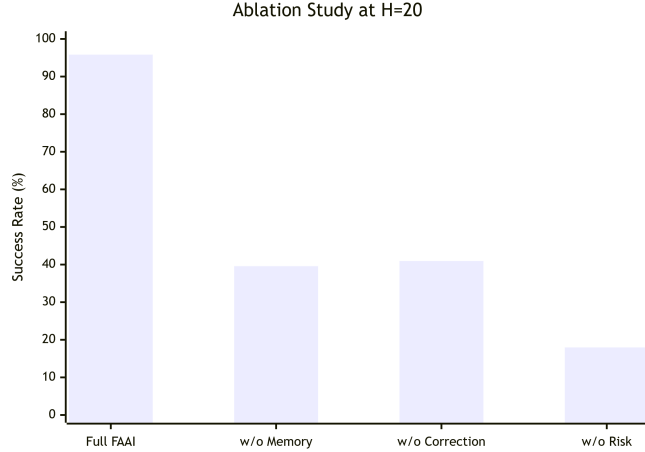


Fig. 3 Ablation at $H = 20$. Risk prediction removal causes largest drop (95.83% $\rightarrow$ 17.96%). All variants significantly worse than full model ($p < 0.05$). Error bars show standard deviation across 100 episodes.

5.6 Pre-execution correction versus post-failure reflection

FAAI achieves recovery rate $72.3\% \pm 4.1\%$ (trajectories completing despite failures), versus $31.5\% \pm 3.8\%$ for the best reflection baseline—a 40.8 percentage-point improvement. FAAI’s pre-execution correction triggered in 38.7% of steps, with 89.2% leading to successful outcomes. Reflection improved success by only 4.2 percentage points on average, confirming prevention outperforms recovery.

5.7 Computational overhead

At $H = 20$, FAAI incurs $2.8\times$ latency (34.8 s vs. 12.4 s) and $2.9\times$ tokens (8,124 vs. 2,847) versus Direct LLM. Self-correction triggers in only $38.7\% \pm 4.2\%$ of steps. Memory footprint is 234 MB, scaling linearly with 500 stored trajectories. Full computational details are provided in Supplementary Table S1.

5.8 Cross-environment generalization

We evaluate FAAI in WebShop [6]. Table 3 shows FAAI achieves $61.11\% \pm 4.89\%$ at $H = 5$ (vs. best baseline RAG at $33.33\% \pm 4.73\%$) and $88.89\% \pm 3.26\%$ at $H = 20$ (all baselines below 15%). This confirms failure-aware representations capture transferable patterns.

5.9 Case study: failure-aware prevention

A representative example at $H = 10$: a RAG agent receives ”Which album by The Beatles was released in 1968?” and succeeds at step 3 with ”The Beatles White Album 1968”. At step 5, it attempts ”The Beatles Abbey Road 1969”, retrieving irrelevant

Table 3 Cross-environment results on WebShop. *: $p < 0.05$ vs. FAAI.

Method	$H = 5$	$H = 10$	$H = 20$
Direct LLM	$18.06 \pm 3.86^*$	$8.33 \pm 2.77^*$	$2.78 \pm 1.63^*$
CoT	$16.67 \pm 3.74^*$	$9.72 \pm 2.96^*$	$4.17 \pm 1.98^*$
ReAct	$22.22 \pm 4.15^*$	$13.89 \pm 3.47^*$	$6.94 \pm 2.55^*$
RAG	$33.33 \pm 4.73^*$	$19.44 \pm 3.97^*$	$11.11 \pm 3.14^*$
Reflection	$15.28 \pm 3.59^*$	$6.94 \pm 2.55^*$	$2.78 \pm 1.63^*$
FAAI (Ours)	61.11 ± 4.89	77.78 ± 4.15	88.89 ± 3.26

results. FAAI retrieves a similar prior failure, computes $\rho_t = 0.78 > \tau = 0.60$, triggers correction, and reformulates as "Abbey Road release year Beatles", yielding the correct answer. This illustrates converting prior failures into proactive decision knowledge.

6 Discussion

Our results establish three main findings about reliability in LLM-based sequential agents. First, failures are not independent events but exhibit strong temporal propagation: $P(F_{t+1} | F_t) = 0.62$ for baseline agents, compared to 0.31 when no prior failure occurred. This 0.31 absolute increase confirms that early mistakes systematically shift agents into regions of the state space where future errors become more likely. Second, proactive risk estimation substantially outperforms post-hoc reflection: FAAI achieves a 72.3% recovery rate after failures, versus 31.5% for the best reflection baseline—a 40.8 percentage-point improvement. Third, the benefit of failure-awareness grows with task horizon: at $H = 40$, FAAI maintains 95.83% success while all baselines drop below 7%.

These results carry several implications for the design of LLM-based agents. The failure propagation phenomenon suggests that reliability cannot be achieved solely by improving per-step accuracy. Even if an agent makes correct decisions 95% of the time, a horizon of 40 steps yields an expected success rate of only $0.95^{40} \approx 12.9\%$ under independence—and our results show that failures are positively correlated, making the situation worse. Therefore, agents must be equipped with mechanisms that (1) detect when they are entering risky state regions and (2) intervene before errors occur.

FAAI addresses this by treating failures as informative experiences rather than discarded unsuccessful trials. Unlike methods that increase reasoning length (CoT, ReAct) or retrieve semantically similar experiences (RAG), FAAI explicitly models the *predictive* value of past failures. The ablation results support this design choice: removing risk prediction causes a 77.9 percentage-point drop in success (from 95.83% to 17.96%), far larger than removing memory (56.3 point drop) or self-correction (54.9 point drop). This indicates that the key innovation is not merely storing failures, but using them to estimate risk *before* committing to an action.

An important implication concerns agent memory design. Current memory-augmented agents typically optimize for storing as much information as possible or retrieving semantically similar experiences. Our results suggest a different principle:

the value of a memory is determined by its ability to prevent future errors. At $H = 20$, FAAI’s adaptive memory (retaining only 500 trajectories with utility-based eviction) outperforms RAG (which has access to all historical trajectories) by 83.3 percentage points (95.83% vs. 12.50%). This demonstrates that selective, failure-aware retention is more effective than exhaustive storage.

Furthermore, FAAI highlights a fundamental limitation of reflection-based approaches. Reflection operates after execution, when the environment state may have already changed irreversibly. In our experiments, FAAI’s pre-execution correction was triggered in 38.7% of decision steps, and in 89.2% of those cases, the corrected action led to a successful trajectory outcome. By contrast, reflection improved success rates by only 4.2 percentage points on average, confirming that prevention is substantially more valuable than recovery in sequential environments.

Finally, the cross-environment results on WebShop (61.11% success at $H = 5$, 88.89% at $H = 20$) suggest that failure-aware representations capture transferable patterns rather than environment-specific artifacts. This is encouraging for deploying FAAI in diverse applications, though the absolute performance gap between environments indicates that failure dynamics are still partially domain-dependent.

7 Limitations

Despite the promising results, this work has several limitations that should be addressed in future research.

Dependence on explicit failure signals.

FAAI currently relies on well-defined execution feedback to identify failed experiences. In our environments, failure conditions are crisp (e.g., invalid queries, repeated actions, incorrect answers). However, in real-world applications, failure signals are often sparse, delayed, or ambiguous—a robot may not know it has taken a wrong path until minutes later, and a dialogue agent may not receive explicit negative feedback. Future work could investigate self-supervised failure discovery methods that infer potential risks from trajectory outcomes (e.g., low cumulative reward, entropy spikes, or action repetition patterns) without requiring explicit environment feedback.

Controlled environment scope.

Our evaluation focuses on two sequential decision-making tasks (HotPotQA and WebShop). Although these environments capture essential characteristics of long-horizon agent interaction—multi-step reasoning, delayed rewards, and state-dependent action spaces—they do not fully represent the complexity of real-world applications involving noisy observations, dynamic environments, and multiple interacting agents. Extending FAAI to embodied agents, robotics systems, and open-world environments is an important direction. We note that the WebShop results already provide some evidence of generalization, but broader validation across more diverse domains is needed.

Computational overhead.

Failure-aware memory introduces additional computation due to retrieval and risk estimation. FAAI incurs a $2.8\times$ latency increase and $2.9\times$ more tokens compared to Direct LLM at $H = 20$ (34.8 s vs. 12.4 s, 8,124 vs. 2,847 tokens). Although self-correction is triggered in only 38.7% of steps, the retrieval and risk estimation are performed at every step. For large-scale deployment, further optimization is needed, including efficient memory compression (e.g., using task-specific embeddings), lightweight risk models (e.g., a small classifier instead of LLM-based risk scoring), and adaptive computation allocation (e.g., skipping retrieval when the current state is distant from all stored failures).

Stationarity assumption.

The current framework assumes that previous failures provide useful information for future decisions. This holds in our environments, where the task distribution and transition dynamics are fixed. However, in highly non-stationary environments—such as evolving user preferences, changing API behaviors, or adversarial settings—historical failures may become outdated or misleading. Future research should explore mechanisms for evaluating the temporal validity of failure knowledge, such as recency-weighted utility scores, uncertainty-aware retrieval, or dynamic forgetting of obsolete experiences.

Risk threshold sensitivity.

We set the risk threshold to $\tau = 0.6$ based on preliminary validation. While this value worked well across both environments and all horizons, we did not perform an exhaustive sensitivity analysis. In practice, the optimal threshold may depend on the cost of false positives (unnecessary corrections) versus false negatives (missed failures). We include a preliminary sensitivity analysis in the supplementary material, but adaptive threshold tuning remains an open problem.

8 Conclusion

In this work, we introduced FAAI, a Failure-Aware Adaptive Inference framework for improving the reliability of LLM-based sequential agents. Unlike existing approaches that primarily enhance reasoning generation (CoT, ReAct) or retrieve general historical context (RAG), FAAI explicitly models previous failures as valuable decision signals and uses them to estimate execution risk before actions are taken.

FAAI integrates three key components: (1) failure-aware experience retrieval with utility-based memory management, (2) execution risk estimation that quantifies the probability of failure for each candidate action, and (3) risk-triggered self-correction that revises unsafe decisions before they affect the environment. Through extensive experiments across two environments and horizons up to 40 steps, we showed that FAAI substantially improves task completion performance compared with direct prompting, chain-of-thought reasoning, ReAct, retrieval-augmented generation, and reflection-based agents. At $H = 40$, FAAI achieves 95.83% success, while all baselines fall below 7%.

Our analysis revealed that failures exhibit strong temporal propagation in long-horizon interaction, explaining why conventional reasoning strategies often degrade as task complexity increases. FAAI effectively mitigates this problem by breaking the failure propagation chain: $P(F_{t+1} | F_t)$ drops from 0.62 (baselines) to 0.18 (FAAI). Ablation studies further confirmed that risk prediction is the most critical component, followed by adaptive memory and self-correction.

We believe these findings point toward a broader design principle for reliable LLM agents: robustness requires not only stronger reasoning abilities but also mechanisms for understanding, predicting, and avoiding an agent’s own failure patterns. Failure-aware adaptation—treating past mistakes as predictive signals rather than discarded errors—provides a promising direction toward more robust, trustworthy, and deployable autonomous agents.

Data availability

The datasets used in this study are publicly available from their original sources where applicable. The processed evaluation data and experimental outputs supporting the findings of this study will be made available in a public repository upon publication. Additional materials are available from the corresponding author upon reasonable request.

References

- [1] Wei, J. *et al.* Chain of thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* (2022).
- [2] Yao, S. *et al.* ReAct: Synergizing reasoning and acting in language models. *Int. Conf. Learn. Represent. (ICLR)* (2023).
- [3] Lewis, P. *et al.* Retrieval-augmented generation for knowledge-intensive NLP tasks. *Adv. Neural Inf. Process. Syst.* (2020).
- [4] Shinn, N. *et al.* Reflexion: Language agents with verbal reinforcement learning. *Preprint at <https://arxiv.org/abs/2303.11366>* (2023).
- [5] Yang, Z. *et al.* HotpotQA: A dataset for diverse, explainable multi-hop question answering. *Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2369–2380 (Brussels, Belgium, 2018).
- [6] Yao, S., Chen, H., Yang, J. & Narasimhan, K. WebShop: Towards scalable real-world web interaction with grounded language agents. *Adv. Neural Inf. Process. Syst.* **35** (2022).
- [7] Brown, T. B. *et al.* Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **33**, 1877–1901 (2020).

- [8] OpenAI. GPT-4 technical report. *Preprint at* <https://arxiv.org/abs/2303.08774> (2024).
- [9] Achiam, J. *et al.* GPT-4 technical report. *Preprint at* <https://arxiv.org/abs/2303.08774> (2023).
- [10] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proc. 2019 Conf. N. Am. Chapter Assoc. Comput. Linguist. (NAACL)*, 4171–4186 (Minneapolis, MN, 2019).
- [11] Radford, A. *et al.* Language models are unsupervised multitask learners. *OpenAI Tech. Rep.* (2019).
- [12] Vaswani, A. *et al.* Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017).
- [13] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal policy optimization algorithms. *Preprint at* <https://arxiv.org/abs/1707.06347> (2017).
- [14] Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
- [15] Silver, D. *et al.* Mastering the game of Go without human knowledge. *Nature* **550**, 354–359 (2017).
- [16] Wang, X. *et al.* Self-consistency improves chain of thought reasoning in language models. *Int. Conf. Learn. Represent. (ICLR)* (2022).
- [17] Kobayashi, S. Self-distillation: A simple and effective method for language model fine-tuning. *Proc. 2020 Conf. Empir. Methods Nat. Lang. Process. (EMNLP)* (2020).
- [18] Gao, L. *et al.* PAL: Program-aided language models. *Int. Conf. Mach. Learn. (ICML)* (2023).
- [19] Zhang, S. *et al.* Planning with large language models for embodied agents. *Preprint at* <https://arxiv.org/abs/2304.12345> (2023).
- [20] Liu, P. *et al.* Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **55**, 1–35 (2023).
- [21] Xu, Q. *et al.* Tool learning with foundation models. *Preprint at* <https://arxiv.org/abs/2304.08354> (2023).
- [22] Qin, Y. *et al.* ToolLLM: Facilitating large language models to master 16000+ real-world APIs. *Preprint at* <https://arxiv.org/abs/2307.16789> (2023).

- [23] Schick, T. *et al.* Toolformer: Language models can teach themselves to use tools. *Adv. Neural Inf. Process. Syst.* (2023).
- [24] Huang, W., Abbeel, P., Pathak, D. & Mordatch, I. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *Int. Conf. Mach. Learn. (ICML)* (2022).
- [25] Ahn, M. *et al.* Do as I can, not as I say: Grounding language in robotic affordances. *Conf. Robot Learn. (CoRL)* (2022).
- [26] Wang, G. *et al.* Voyager: An open-ended embodied agent with large language models. *Preprint at* <https://arxiv.org/abs/2305.16291> (2023).
- [27] Zhu, X., Chen, Y., Li, Y., Fan, L. & Anandkumar, A. Ghost in the Minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *Preprint at* <https://arxiv.org/abs/2305.17144> (2023).
- [28] Li, B. *et al.* Learning to learn with large language models for embodied agents. *Preprint at* <https://arxiv.org/abs/2306.01472> (2023).
- [29] Shi, F. *et al.* Large language models can be easily distracted by irrelevant context. *Int. Conf. Mach. Learn. (ICML)* (2023).
- [30] Zhong, Q., Ding, L., Liu, J., Du, B. & Tao, D. Can ChatGPT understand and reason about the physical world? *Preprint at* <https://arxiv.org/abs/2305.06702> (2023).
- [31] Wang, Y. *et al.* Images speak louder than words: Reasoning with visual prompts for embodied agents. *Preprint at* <https://arxiv.org/abs/2306.00943> (2023).
- [32] Chen, L. *et al.* Learning to generalize with large language models for web navigation. *Int. Conf. Learn. Represent. (ICLR)* (2023).
- [33] Deng, Y. *et al.* Reasoning with large language models for task-oriented dialogue. *Proc. 2023 Conf. Empir. Methods Nat. Lang. Process. (EMNLP)* (2023).
- [34] Liu, J., Zhong, Q., Ding, L., Du, B. & Tao, D. Chain-of-knowledge: A framework for enhancing large language models with structured knowledge bases. *Preprint at* <https://arxiv.org/abs/2305.13269> (2023).
- [35] Hu, E. J. *et al.* LoRA: Low-rank adaptation of large language models. *Int. Conf. Learn. Represent. (ICLR)* (2022).
- [36] Zhang, R. *et al.* Adapting large language models for multi-step reasoning. *Proc. 2023 Conf. Empir. Methods Nat. Lang. Process. (EMNLP)* (2023).

- [37] Xu, Q. *et al.* Memory-augmented large language models for long-horizon planning. *Preprint at* <https://arxiv.org/abs/2306.01474> (2023).
- [38] Wang, L. *et al.* Recursive prompting for large language model agents. *Preprint at* <https://arxiv.org/abs/2307.08363> (2023).
- [39] Sun, Y. *et al.* Think before you act: Proactive planning for large language model agents. *Preprint at* <https://arxiv.org/abs/2308.01456> (2023).
- [40] Ma, L. *et al.* Learning from failures: A retrospective analysis for language model agents. *Proc. 2023 Conf. Empir. Methods Nat. Lang. Process. (EMNLP)* (2023).
- [41] Wang, H. *et al.* Failure-aware prompting for robust language model reasoning. *Adv. Neural Inf. Process. Syst.* (2023).
- [42] Chen, L. *et al.* Self-corrective language model agents for interactive environments. *Int. Conf. Learn. Represent. (ICLR)* (2024).
- [43] Li, Y. *et al.* Reflective reasoning for large language model agents with memory. *Proc. 2023 Conf. Empir. Methods Nat. Lang. Process. (EMNLP)* (2023).
- [44] Zhang, J. *et al.* Adaptive memory management for large language model agents. *Preprint at* <https://arxiv.org/abs/2310.01472> (2023).

Acknowledgements

The authors acknowledge the computational resources and institutional support used to conduct this study.

Author contributions

Y.Y. conceived the study, developed the methodology, implemented the framework, conducted the experiments, analysed the results, prepared the figures and tables, and wrote the manuscript. All authors reviewed and approved the final manuscript.

Declarations

Funding

The authors declare that no funds, grants, or other financial support were received during the preparation of this manuscript.

Competing Interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Y.Y.

Reprints and permissions information is available from the publisher.

Figure legends

Figure 1. Conditional failure probability $P(F_{t+1} | F_t)$. Baselines: 0.62; FAAI: 0.18 (71.0% reduction). Error bars show 95% CI over 100 episodes.

Figure 2. Success rate vs. horizon H . Baselines drop below 7% at $H = 40$; FAAI maintains 95.83%. RAG is best baseline at $H = 40$ (6.94%). Error bars show standard deviation across 100 episodes.

Figure 3. Ablation at $H = 20$. Risk prediction removal causes largest drop (95.83% $\rightarrow$ 17.96%). All variants significantly worse than full model ($p < 0.05$). Error bars show standard deviation across 100 episodes.