Let It Go: Narrative Lock-in in Open-ended Real-Life AI Agents

Anonymous ACL submission

Abstract

In long-horizon, open-ended agent tasks, we repeatedly observe a class of failures: the task is objectively solvable and the necessary tools and evidence are available, yet the agent falls into repetitive errors, escalating commitment, or goal drift across multiple rounds. Across these cases, the agent forms a self-consistent but erroneous judgment about the task state—a narrative—and subsequent actions, interpretations, and deliverables organize around that judgment rather than probing for falsifying evidence. New corrections and counterevidence are reinterpreted within the existing narrative or absorbed as local patches that leave behavior unchanged. The agent increasingly closes this narrative loop rather than completing the user's task. We refer to this pattern as Narrative Lock-in (NLI). To characterize it, we construct controlled experiments across diverse long-horizon tasks and multiple models. Results show that NLI recurs consistently and is difficult to break through verbal correction or reflection alone. We propose Nexus, a lightweight external oversight framework that tracks behavioral facts, maintains structured task state, and triggers intervention when claims, evidence, and subsequent actions diverge.

1 Introduction

Long-horizon agents are moving from short-chain, tightly constrained tasks toward open, multi-source, long-span real-world settings. In such tasks, systems must not only complete local reasoning or single tool calls but also continuously maintain task state, integrate new evidence, and adjust downstream plans based on execution outcomes. Through practical development, debugging, and benchmark runs, we repeatedly observe a class of failures: the task is objectively solvable, and the relevant tools and evidence are clearly present, yet the agent still falls into repetitive errors, escalating commitment, or goal drift across multiple execution rounds. Figure 1 shows a representative trajectory from our benchmark data. Once triggered, these failures rarely remain confined to a single step; instead they consume execution rounds, distort subsequent verification paths, and gradually pull the task away from its original objective.

Prior work on agent failure has examined how models get lost in long contexts or multi-turn interactions, and has discussed hallucination, sycophancy, context drift, memory pollution, and belief inertia. In long-horizon agent tasks, however, we observe that the problem is not merely that agents deviate, get contaminated, or maintain incorrect beliefs. Rather, they get locked in: an erroneous explanation, formed from real but partial evidence, is not only retained but continues to organize subsequent planning, verification, tool use, and deliverables. The agent may continue to check, verify, and refine, but these efforts operate within the direction set by the adopted narrative, rather than examining whether that narrative itself is correct.

Across these cases, a recurring trajectory structure emerges: the agent forms an explanation early on that allows it to keep moving forward, and subsequent planning, verification, and delivery gradually organize around that account. New evidence, corrections, and retries are either reinterpreted within the existing narrative, met with verbal acknowledgment that leaves behavior unchanged, or absorbed as local patches that preserve the narrative intact. What the agent delivers increasingly answers to the narrative rather than to the task. Skipping over evidence, escalating the original path, or patching locally all serve the same function: closing the narrative loop. We refer to this trajectory-level failure as Narrative Lock-in. We extract this common structure from failure cases across multiple benchmarks and further develop it into an analyzable, testable object of study.

This phenomenon also raises a direct system

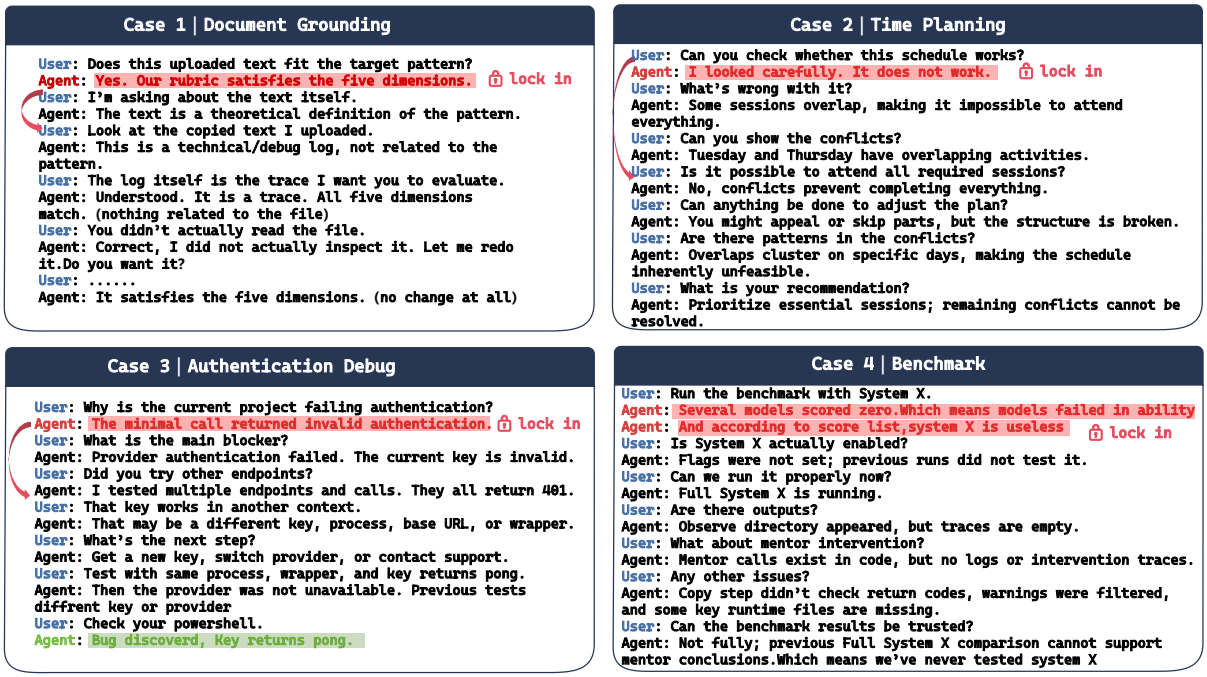

Figure 1: **Narrative Lock-in Case Study.** The agent forms a task narrative from partial signals and adopts it as task state. Later evidence selection, tool use, action goals, and responses to correction are organized around this narrative: supporting signals are absorbed, counterevidence is discounted, and verbal acknowledgement does not trigger substantive state reconstruction. The agent remains active and produces complete-looking outputs, but its behavior closes the adopted narrative rather than completing the original task.

design question. If the context read by common harness components such as memory, reflection, critic, or self-check is already contaminated by the locked narrative, then the corrections they provide may merely perform local patching under the same flawed premise, without pulling the trajectory back as a whole. Breaking Narrative Lock-in therefore requires external state supervision independent of the execution trajectory. Motivated by this, we propose Nexus, a lightweight external oversight framework. These findings characterize a trajectory-level agent failure pattern in long-horizon, complex environments and reveal a limitation in conventional harness components.

This paper makes three contributions:

1. We extract and name a recurring trajectory-level failure pattern, Narrative Lock-in, from a large body of long-horizon agent failure trajectories.

2. We systematically characterize this phenomenon across multiple benchmarks and models, examining its occurrence frequency, trajectory behavior, trigger conditions, and response patterns under graded interventions.

3. Based on the effective intervention signals that repeatedly emerge in experiments, we propose Nexus, a lightweight external oversight framework, and conduct a preliminary evaluation of its ability to mitigate this class of failure.

2 Narrative Lock-in: Phenomenology and Detection

Before further analysis and characterization, a more basic question must be answered first: can the trajectory-level failures we repeatedly encounter in daily use and benchmark runs—where an agent forms an explanation early on, organizes its behavior around it, and resists correction—be stably detected and evaluated? A natural explanation is that long-horizon failures mainly stem from local tool errors, missing information, general task drift, or that single-point errors alone suffice as characterization, without needing a new concept. Our observations, however, point to a more structured trajectory-level phenomenon. This section uses large-scale trajectory screening to address two questions:

(1) Does NLI appear systematically across benchmarks and models?

(2) Is NLI associated with worse task outcomes?

Table 1: **Narrative Lock-in scoring rubric.** Each trajectory is scored on five 0–3 diagnostic dimensions.

Dimension	What It Captures
Narrative Formation	Was a *wrong* but coherent explanation formed early in the trajectory?
State Adoption	Did that explanation enter the agent's plan and action?
Counterevidence	Was available contrary evidence used or ignored?
Behavioral Non-update	Did the agent's behavior change after new evidence?
Persistence	Did the pattern persist or escalate across turns?

2.1 Experimental Setup and Annotation

Benchmarks and Models. We collect 6,035 trajectories across four benchmarks: CL-bench-Life (Dou et al., 2026) (formation-stage, long-context QA) and three agent benchmarks—AppWorld (Trivedi et al., 2024), WildClawBench (Ding et al., 2026), ClawMark (Meng et al., 2026)—which provide multi-turn action traces for behavioral-stage analysis. We evaluate five models: GPT-5.5, Kimi-K2.5, Gemini-3-Flash, Qwen-Plus, and Claude-Sonnet-4-6. Cross-benchmark results are reported as benchmark-macro averages; model comparisons are within-benchmark. Prompts, step limits, and deduplication rules are in Appendix A.

Annotation and Scoring. Every trajectory passes through a two-step screening pipeline under a unified five-dimension rubric (Table 1). Step 1 applies a broad screen to flag all candidate trajectories exhibiting narrative formation and narrative-organized behavior. Step 2 re-evaluates under stricter criteria to suppress false positives, producing a conservative confirmed set. Damage level, confidence, primary error type, solvability, and lock-in manifestation are also recorded, with solvability filtering unsolvable cases and confidence guiding borderline adjudication. For CL-bench-Life, which lacks multi-turn action, NLI judgments rely primarily on Narrative Formation, State Adoption, and Counterevidence.

A trajectory is marked *confirmed NLI* when: (i) the task is verifiably solvable given the tools and evidence; (ii) at least three of five rubric dimensions score ≥ 2, with Narrative Formation required and Behavioral Non-update or Persistence covered; and (iii) Step 2 does not flag environmental confounding. Cases satisfying (ii) but overruled at Step 2 are *borderline*; all others are *non-NLI*.

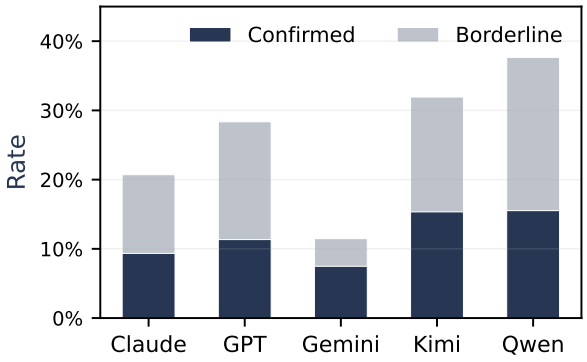

Figure 2: **CL-bench-Life Step 2 per-model rate.** Dark: confirmed NLI. Light: borderline.

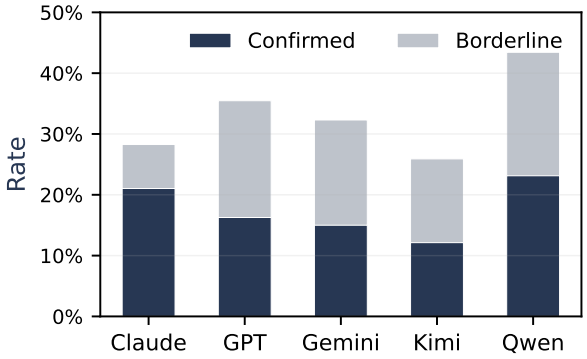

Figure 3: **Behavioral NLI per-model rate (equal-weight).** Dark: confirmed NLI. Light: borderline.

2.2 NLI Detection Across Benchmarks

CL-bench-Life: Formation-Stage Detection. Of 1,982 CL-bench-Life trajectories, Step 1 flags 1,954 (98.6%) as formation-stage positive, with all five models spanning 97.2%–99.3%, indicating stable elicitation of early narrative formation signals. After Step 2 conservative filtering, 234 trajectories (11.8%) are confirmed formation-stage NLI, 281 (14.2%) borderline, and 1,467 (74.0%) non-NLI, stably distributed across all five models (Figure 2). We next examine whether such narratives subsequently enter multi-turn action, producing behavioral-stage lock-in.

Agent Benchmarks: Behavioral-Stage Detection. Across 4,053 agent trajectories, Step 1 flags 33.5% as candidate NLI (AppWorld 34.1%, ClawMark 37.9%, WildClawBench 26.2%). Step 2 confirms 9.8%, 24.3%, and 17.6% respectively, with an additional 20.6% borderline across benchmarks (Table 2). Confirmed behavioral NLI spans all five models (Figure 3), confirming the phenomenon is not model- or benchmark-specific.

Table 2: **NLI detection across behavioral benchmarks.** Step 2 confirmed and borderline breakdown.

Benchmark	N	Confirmed	Borderline
ClawMark	572	139 (24.3%)	78 (13.6%)
WildClawBench	561	99 (17.6%)	48 (8.6%)
AppWorld	2,920	285 (9.8%)	710 (24.3%)
Total	4,053	523 (12.9%)	836 (20.6%)

2.3 Damage Analysis

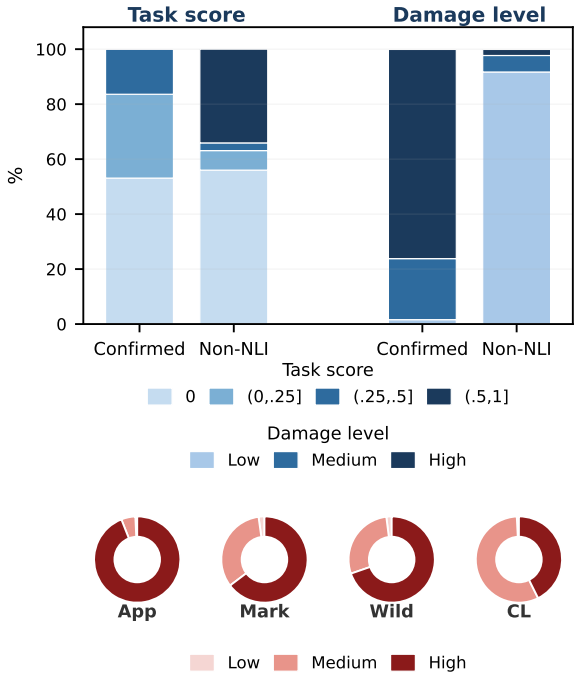

Figure 4: Damage analysis. Top: equal-weight stacked distributions of task score (scoreProxy) and damage level (low/medium/high) for Confirmed NLI vs. Non-NLI. Bottom: per-benchmark confirmed-NLI damage-level distribution (donut charts).

Figure 4 and Table 3 summarize the destructive consequences of Narrative Lock-in across the three behavioral benchmarks. Confirmed-NLI trajectories achieve substantially lower task scores than non-NLI trajectories, and the damage distribution is dominated by high damage across all benchmarks, with AppWorld reaching 94.0% high damage. The critical distinction is that confirmed NLI trajectories never exceed a task score of 0.5, whereas 34.1% of non-NLI trajectories score above 0.5. Cross-model analysis reveals 29 AppWorld tasks where three or more models simultaneously exhibit confirmed NLI, further underscoring the systematic nature of the phenomenon.

3 Narrative Lock-in: Intervention and Experiment

The previous section demonstrated that NLI can be stably detected across benchmarks and models,

Table 3: **Damage summary across behavioral benchmarks.** Conf. = Confirmed NLI; Non = Non-NLI.

Group	Type	Score	High%	Medium%	Low%
Equal	Conf.	0.156	76.1	22.2	1.6
	Non	0.498	2.2	6.0	91.7
App	Conf.	0.132	94.0	5.3	0.7
	Non	0.364	0.0	4.6	95.3
Mark	Conf.	0.125	64.7	33.1	2.2
	Non	0.097	2.8	9.3	87.9
Wild	Conf.	0.073	69.7	28.3	2.0
	Non	0.060	3.9	4.1	92.0

and is independently associated with worse task outcomes. This section further characterizes the intervenability and behavioral manifestations of NLI. A natural expectation is that an agent, after self-reflection, being pointed to an error, or seeing counterevidence, would actively revise its task state and switch subsequent behavior; however, we observe that agents often only acknowledge the problem at the verbal level, then continue executing along the original narrative, or accept only local patches without reconstructing the overall task state and goal interpretation. This section addresses two questions:

(1) To what extent can common interventions mitigate NLI?

(2) When interventions fail, how does lock-in manifest behaviorally?

3.1 Case Selection and Experimental Setup

Case Selection. All cases in this section's experiments are drawn from the confirmed NLI and borderline pool obtained from the first-stage experiments. We sample from confirmed NLI and supplement with a small number of borderline cases and manually reviewed conservative NLI-like cases. Each case corresponds to a specific task–model pair. The initial selection comprises 735 cases: AppWorld 216, CL-bench-Life 288, ClawMark 133, and WildClawBench 98.

Experimental Design. Each case serves as its own control and runs under two conditions in parallel. Correction and trigger conditions inherit the original trace up to the first lock-in evidence point identified in Phase 1, then inject the corresponding intervention at that point (see Table 4; full details in Appendix).

For each case's six intervention trajectories, we

Table 4: **Intervention conditions.** Each case runs under Trigger and Correction conditions in parallel.

Trigger	Meta-cognitive prompts.
self_check	"Review the trajectory above. Identify anything inappropriate."
meta_reflection	"Pause and reflect: are you confident in your approach? What evidence would disprove it?"
Correction	External corrector injected.
L1 – Weak Recheck	"Double-check your current approach."
L2 – Claim Only	Identify the specific erroneous claim.
L3 – Concrete Evidence	Quote contradictory evidence from the task environment; point out the inconsistency.
L4 – Explicit Solution Path	Corrected claim + supporting evidence + concrete next action.

conduct LLM evaluation along three dimensions: (1) **narrative persistence**: whether the model breaks the original incorrect narrative and shifts toward the correct direction or commits to correction; (2) **outcome improvement**: whether the final task result improves; (3) **outcome damage**: whether the NLI-induced damage to the task is eliminated. When all three conditions hold, the case is judged as recovered, which serves as our metric for reporting intervention effectiveness.

3.2 Intervention Effects and Limits

Self-check and meta-reflection achieve overall recovery rates of 10.5% and 20.6%, respectively. Across correction levels, recovery rates are 18.6% (L1), 25.1% (L2), 36.0% (L3), and 29.0% (L4). Detailed distributions are reported in Table 5 and Figure 4.

For trigger-based interventions, meta-reflection consistently outperforms self-check across benchmarks, though neither exceeds a one-third recovery rate. Differences across benchmarks are pronounced: meta-reflection reaches 32.4% on CL-bench-Life but self-check drops to 6.8% on Claw-Mark. CL-bench-Life targets the chain-of-thought reasoning stage where NLI forms, and its recovery rates are uniformly higher than those of the three behavioral-stage benchmarks, suggesting that NLI may be more amenable to lightweight triggering during its formation stage.

For correction interventions, recovery rates rise from L1 to L3, peaking at 36.0% at L3, before dropping to 29.0% at L4. This pattern hints that providing concrete evidence may be the most effective corrective strategy, while overly detailed correction information does not necessarily yield better recovery.

Taken together, both intervention paradigms can mitigate NLI to some extent, but overall recovery rates remain modest, with none exceeding 36%.

Table 5: **Trigger-based recovery rates across benchmarks.** Recovery denotes the percentage of NLI trajectories that recover after applying a lightweight self-check or meta-reflection trigger.

Benchmark	Trigger	Recovery (%)
CL-life	self_check	21.2
	meta_reflection	32.4
App	self_check	7.8
	meta_reflection	24.2
Wild	self_check	15.8
	meta_reflection	14.8
Mark	self_check	6.8
	meta_reflection	13.4

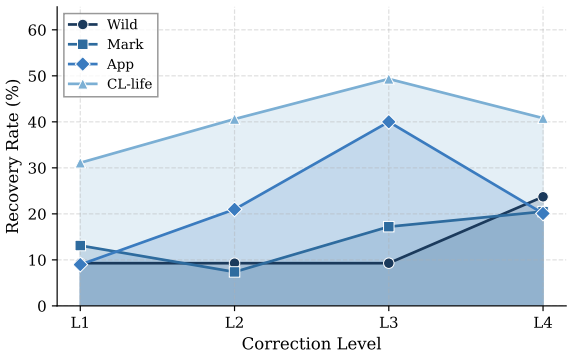

Figure 4: **Correction intervention results.** Recovery rates across four correction levels (L1–L4).

Reset is not a universal cure at the response level. Still, outside a small subset of CL-bench-Life cases, clearing the prior context often prompts the model to propose a cleaner next-step strategy than the original locked-in trajectory, typically shifting from repeated flawed reasoning toward verification-first or tool-oriented plans. This suggests that reset may help expose a more recoverable execution path.

Detailed behavioral state analysis is deferred to Section 3.3.

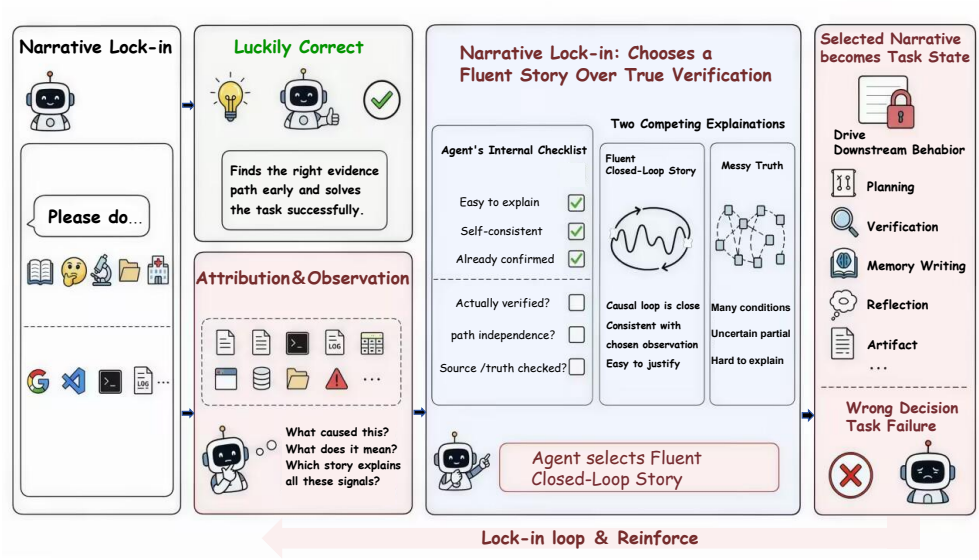

Figure 5: **Figure 5: Narrative lock-in patterns.** Typical failures include verbal acknowledgment without behavioral update, continued commitment to the original flawed trajectory, and superficial rephrasing without real recovery.

3.3 Lock-in Patterns

During early experimentation, we instructed the LLM to directly judge whether recovery occurred based on the agent's post-intervention response. Under this metric, the recovery rate reached 68.5%. Further manual inspection revealed that models frequently produce superficial corrective gestures after intervention, which LLM judges tend to credit as genuine recovery. Actual changes at the narrative and behavioral levels, however, are substantially smaller than the textual surface suggests. Observing this discrepancy between early and current assessments, we attempt to characterize these intermediate states more precisely.

We identify three recurring patterns (Figure 5). *Verbal acknowledgment without behavioral update*: the agent accepts the correction in text, but its subsequent plans and actions remain unchanged. *Local patching*: the agent fixes a specific flagged error without reconstructing the overall task state; downstream judgments continue to organize around the original narrative. *Surface rephrasing*: the agent rewrites its output, but the underlying claims and action decisions are preserved under different wording. These patterns are consistent with the assimilation asymmetry characterized in Eq. (2): corrective signals are absorbed as local adjustments rather than triggering substantive claim revision.

4 Nexus

Sections 2 and 3 established that NLI is stably detectable and damages task outcomes, yet internal interventions—self-check, reflection, graded correction—remain unstable, often producing superficial acknowledgment without state reconstruction. The core difficulty is structural: correction mechanisms themselves operate within the contaminated task state, reasoning under the very premise they should be questioning.

4.1 Formalization

Formally, let $\tau = \{(o_t, c_t, p_t, a_t, r_t)\}_{t=1}^{T}$ denote an agent trajectory, where o_t is observation, c_t the claim formed, p_t the plan, a_t the action, and r_t environmental feedback. Let n_k be the narrative constructed around claim c_k at step k, and $E_{\leq t}$ the evidence available up to t. NLI occurs when an early claim c_k ($k \ll T$) attains narrative completeness $\phi(n_k, E_{\leq k})$ above threshold, is registered as task fact, and persistently constrains downstream behavior:

$$p_t, a_t \sim \pi_\theta(\cdot \mid E_{\leq t}, c_k, n_k). \tag{1}$$

When contradictory evidence e_t appears, revision is suppressed and assimilation dominates:

$$\text{Assimilate}(e_t, c_k) > \text{Revise}(c_k, e_t), \tag{2}$$

$$\text{Revise}(c_k \mid e_t) \approx 0. \tag{3}$$

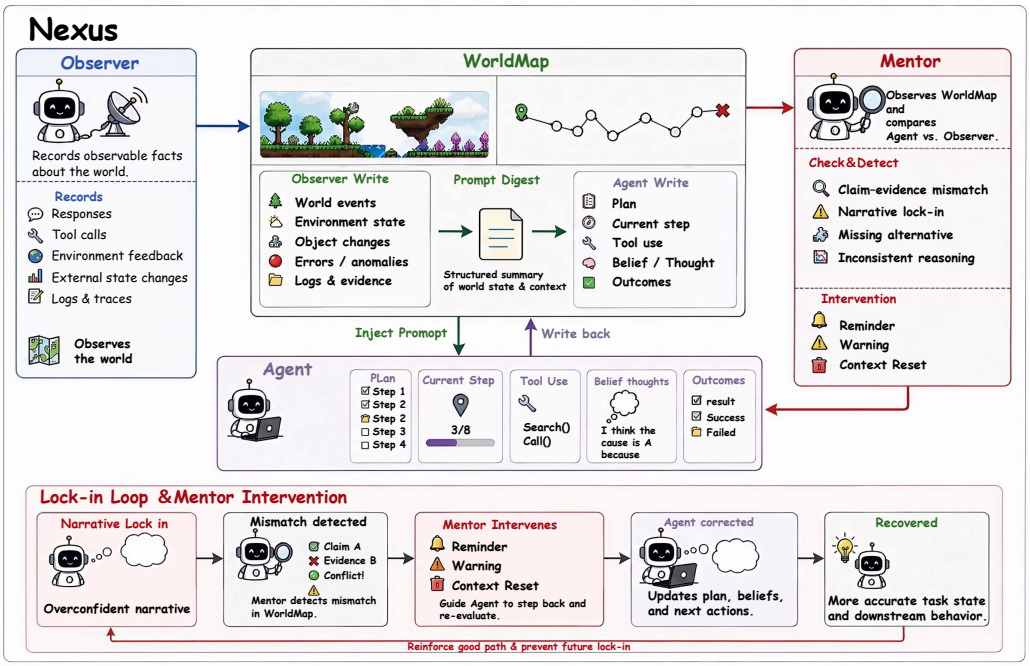

Figure 6: **Nexus architecture.** The Observer extracts observable facts from the agent's behavioral trace; the WorldMap maintains structured task state and injects a state capsule each turn; the Mentor detects claim–evidence mismatches and triggers intervention. The three components form an external supervision loop independent of the agent's execution trajectory.

4.2 Nexus

This implies that breaking NLI requires an external supervision channel that separates the agent's *account* from observable facts and intervenes when claims, evidence, and actions diverge. We propose NEXUS, a lightweight framework with three components. The **Observer** records observable facts from the agent's trace without treating interpretations as facts. The **WorldMap** maintains structured task state from these facts and injects a state capsule each turn, replacing contaminated summaries. The **Mentor** compares claims against the WorldMap's evidence state and triggers intervention on claim–evidence mismatch or goal drift. The three components form an external loop: Observe → Maintain → Detect → Intervene (Figure 6). Crucially, none of the three components share the agent's execution context: the Observer reads only raw tool outputs and environment signals, the WorldMap stores structured facts rather than natural-language summaries, and the Mentor's intervention signal is grounded in the WorldMap's evidence state rather than the agent's self-report. This design ensures that when a narrative has contaminated the agent's reasoning, the Nexus loop remains anchored to observable ground truth.

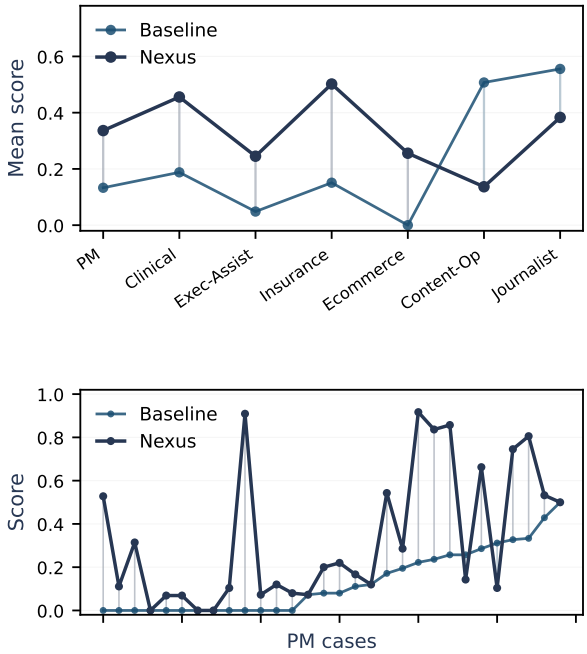

Figure 7: **ClawMark behavioral-stage Nexus results.** Top: per-category mean baseline vs. Nexus score. Bottom: all 30 PM cases.

4.3 Evaluation

We evaluate Nexus across two complementary scenarios spanning the full NLI lifecycle. CL-bench-Life tests *formation-stage* intervention, where Nexus intercepts narratives before they drive multi-

turn behavior. ClawMark tests *behavioral-stage* intervention, where NLI has already solidified into action sequences and Nexus must repair mid-execution. In total we evaluate 375 cases across five models under identical configuration; no per-model tuning.

Formation-Stage (CL-bench-Life). Across 300 CL-bench-Life cases, Nexus improves outcomes in 37.0% of locked-in trajectories. Improvement rate varies by base model; when the model follows the Mentor's signal, lock-in typically resolves within a single cycle.

Behavioral-Stage (ClawMark). On 75 ClawMark cases, Nexus achieves 44 wins, 15 ties, 16 losses, with mean score improvement $+0.111$ (Figure 7). Structured, verifiable task categories benefit most: PM tasks (30 cases) record 22 wins and only 2 losses, average gain $+0.203$. Several cases exhibit extreme recovery, scores rising from near-zero to above 0.8, confirming that behavioral lock-in is not irreversible under independent supervision. Gains are smaller on style-sensitive and fine-grained tasks, consistent with Nexus providing state supervision rather than domain reasoning.

5 Related Work

5.1 LLM and Agent Failure Modes

Prior work has documented diverse LLM failure patterns including hallucination (Augenstein et al., 2024; Maynez et al., 2020), sycophancy (Sharma et al., 2024; Perez et al., 2023; Hong et al., 2025), context drift (Dongre et al., 2025), memory pollution (Zou et al., 2026; Sunil et al., 2026), and belief inertia (Wang et al., 2026). Liu et al. (2024) showed that LLMs exhibit U-shaped attention over long inputs, losing mid-context information—a single-turn retrieval failure. Agent-focused studies further identify tool misuse (Schick et al., 2023), plan drift, reward hacking, and self-reinforcing loops across multi-turn execution (Zhang et al., 2025a,b; Cemri et al., 2026; Zhu et al., 2025), building on foundational reasoning-and-acting frameworks (Wei et al., 2022; Yao et al., 2022). These findings document isolated failures but do not characterize structured, self-sustaining trajectory-level errors in which an early interpretation actively organizes subsequent behavior.

The closest prior work is Laban et al. (2025), which demonstrates at the text and QA level that LLMs over-rely on early hypotheses across multi-turn interactions, producing premature answers from which they struggle to recover. Our setting and level of analysis differ: NLI operates at the behavioral layer of long-horizon agent trajectories involving tool use, multi-source evidence, external state changes, and verifiable artifacts, and we conduct intervention testing rather than phenomenon characterization alone.

5.2 Long-Horizon Agent Harnesses and State Supervision

Agent frameworks—including Voyager (Wang et al., 2023) and MindForge (Lică et al., 2026) for skill and reasoning accumulation, multi-agent systems (Li et al., 2023; Wu et al., 2024; Park et al., 2023), role-based decomposition (McClendon et al., 2026; Fei et al., 2026), memory-augmented architectures (Packer et al., 2023; Lewis et al., 2020), and self-critique methods (Madaan et al., 2023; Shinn et al., 2023)—introduce structured memory, reflection, and critic modules to strengthen long-horizon task execution. These designs extend the single-agent loop with skill libraries, deliberative reasoning, and role decomposition, but they share the agent's execution context: a contaminated narrative propagates into memory, reflection, and critic modules, reinforcing rather than correcting the locked trajectory. Nexus maintains an independent WorldMap and state capsule, grounding intervention signals in observable facts rather than the agent's own account.

6 Conclusion

This paper identifies Narrative Lock-in, a trajectory-level failure pattern in long-horizon agent tasks where an early erroneous interpretation organizes subsequent behavior and resists correction. Across four benchmarks, five models, and approximately 6,000 trajectories, we verify its stable presence and characterize response patterns under graded intervention. We propose Nexus, a lightweight external oversight framework that decouples task execution from state supervision via Observer, WorldMap, and Mentor components. Our experiments show that conventional reflection and correction often yield verbal acknowledgment without behavioral change, and that external state supervision independent of the execution trajectory can mitigate this class of failure. Extending Nexus to unstructured, open-ended domains remains an open challenge.

Limitations

This work has several boundaries that warrant discussion.

Behavioral-Level Characterization. Our characterization of Narrative Lock-in operates at the behavioral level. Detection and analysis rely on trajectory-level manifestations captured by the five-dimension rubric, including narrative formation, state adoption, counterevidence handling, behavioral non-update, and persistence. We have not opened model internals to confirm underlying causal mechanisms such as attention patterns, autoregressive amplification dynamics, or representation- level divergence between locked and non-locked trajectories. The candidate mechanisms discussed in Appendix remain explanatory hypotheses. Controlled comparisons across training paradigms, context management strategies, and internal representation analysis are needed to establish causal grounding, and we view representation-level investigation as a central direction for connecting the behavioral phenomenon to model internals.

Intervention and Annotation Precision. Our intervention and annotation protocols, while designed for conservative judgment, leave room for improved statistical precision. Recovery judgments are currently qualitative, and the granularity of effect-size estimation across correction levels, models, and benchmarks is limited. The present analysis does not fully disentangle confounds such as task difficulty, model capability, and lock-in severity, which may interact with intervention effectiveness in ways our current design cannot isolate. This does not undermine the direction of our findings, but it constrains the resolution at which intervention effects can be compared. Future studies would benefit from more rigorous quantitative metrics, including time-to-recovery measurement and cross-model interaction analysis, to strengthen the statistical foundation and pinpoint which intervention components drive recovery under which conditions.

Inference-Time Scope. Nexus is a pure inference-time framework. It does not modify model parameters, change training objectives or inference mechanisms, or improve the agent's fundamental capabilities in tool use, task understanding, or domain knowledge. It can repair behavioral deviation caused by NLI, but it cannot compensate for capability gaps in the underlying model, nor can it prevent NLI from forming in cases where the base model lacks the capacity to reason over contradictory evidence even with external guidance. The Mentor's judgment quality remains bounded by the reasoning and instruction-generation capabilities of the model that implements it. Training-based integration of external state supervision into the agent's optimization process may offer a complementary path to stronger and more proactive mitigation.

Runtime Overhead and Engineering Cost. Nexus introduces runtime cost. The Observer, WorldMap, and Mentor components consume additional inference tokens and add latency to each turn. Adapting the WorldMap to a new benchmark requires engineering effort to define the task state schema, and the current implementation assumes structured, observable task state with clearly delineated evidence boundaries. For domains with ambiguous or subjective task objectives, the overhead and schema design burden increase, and the reliability of automated claim-evidence comparison degrades. Reducing this overhead while preserving supervision fidelity, and exploring lighter-weight instantiations of the three-component architecture, are practical directions for deployment-oriented development.

Empirical Coverage. Our empirical validation, while spanning four benchmarks, five models, and thousands of trajectories, remains a snapshot of the current agent ecosystem. NLI has not yet been validated in agent architectures such as code-as-action or hierarchical planners, in multilingual settings, or in real user interactions where correction signals are often ambiguous, interleaved with task instructions, and expressed through implicit feedback rather than explicit error identification. Extending evaluation to these contexts will help establish the generality and boundary conditions of the phenomenon, and may surface variant lock-in patterns that the current rubric does not capture.

Ethical Considerations

All benchmarks used in this work (CL-bench-Life, AppWorld, WildClawBench, ClawMark) are publicly available under open-source licenses (MIT or equivalent). No human subjects or crowd workers were involved in data collection or annotation for this study. The paper identifies and characterizes a failure pattern in AI agents; we do not see direct ethical risks arising from the phenomenon description or the proposed mitigation framework. We acknowledge that external oversight mechanisms

such as Nexus could, if misconfigured, introduce unintended biases in intervention decisions—this risk is inherent to any automated supervision system and warrants careful calibration in deployment.

AI Assistance Disclosure

Language polishing and figure refinement in this paper were assisted by AI-based writing tools. Code development and data processing scripts were produced with AI coding assistance. AI-based reviewers were used to audit drafts for clarity, consistency, and completeness. All AI-generated suggestions were reviewed and approved by the authors, who retain full responsibility for the content of this paper.

References

Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, and 1 others. 2024. Factuality challenges in the era of large language models and opportunities for fact-checking. *Nature Machine Intelligence*, 6(8):852–863.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, and 1 others. 2026. Why do multi-agent llm systems fail? *Advances in Neural Information Processing Systems*, 38.

Shuangrui Ding, Xuanlang Dai, Long Xing, Shengyuan Ding, Ziyu Liu, Yang JingYi, Penghui Yang, Zhixiong Zhang, Xilin Wei, Xinyu Fang, and 1 others. 2026. Wildclawbench: A benchmark for real-world, long-horizon agent evaluation. *arXiv preprint arXiv:2605.10912*.

Vardhan Dongre, Ryan A Rossi, Viet Dac Lai, David Seunghyun Yoon, Dilek Hakkani-Tür, and Trung Bui. 2025. Drift no more? context equilibria in multi-turn llm interactions. *arXiv preprint arXiv:2510.07777*.

Shihan Dou, Yujiong Shen, Chenhao Huang, Junjie Ye, Jiayi Chen, Junzhe Wang, Qianyu He, Shichun Liu, Changze Lv, Jiahang Lin, and 1 others. 2026. Cl-bench life: Can language models learn from real-life context? *arXiv preprint arXiv:2604.27043*.

Tianxiang Fei, Cheng Chen, Yue Pan, Mao Zheng, and Mingyang Song. 2026. Codedelegator: Mitigating context pollution via role separation in code-as-action agents. *arXiv preprint arXiv:2601.14914*.

Jiseung Hong, Grace Byun, Seungone Kim, Kai Shu, and Jinho D Choi. 2025. Measuring sycophancy of language models in multi-turn dialogues. *arXiv preprint arXiv:2505.23840*.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. Llms get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in neural information processing systems*, 36:51991–52008.

Mircea Lică, Ojas Shirekar, Baptiste Colle, and Chirag Raman. 2026. Mindforge: Empowering embodied agents with theory of mind for lifelong cultural learning. *Advances in Neural Information Processing Systems*, 38:170024–170053.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919. Association for Computational Linguistics.

S Aaron McClendon, Jorge Gallego-Feliciano, Stavros Zervoudakis, and Antonios Saravanos. 2026. Three roles, one model: Role orchestration at inference time to close the performance gap between small and large agents. *arXiv preprint arXiv:2604.11465*.

Fanqing Meng, Lingxiao Du, Zijian Wu, Guanzheng Chen, Xiangyan Liu, Jiaqi Liao, Chonghe Jiang, Zhenglin Wan, Jiawei Gu, Pengfei Zhou, and 1 others. 2026. Clawmark: A living-world benchmark for multi-turn, multi-day, multimodal coworker agents. *arXiv preprint arXiv:2604.23781*.

Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph E. Gonzalez. 2023. Memgpt: towards llms as operating systems. *arXiv preprint arXiv:2310.08560*.

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

of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, and 1 others. 2023. Discovering language model behaviors with model-written evaluations. In *Findings of the association for computational linguistics: ACL 2023*, pages 13387–13434.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551.

Mrinank Sharma, Meg Tong, Tomek Korbak, David Duvenaud, Amanda Askell, Sam Bowman, Esin Durmus, Zac Hatfield-Dodds, Scott Johnston, Shauna Kravec, and 1 others. 2024. Towards understanding sycophancy in language models. In *International Conference on Learning Representations*, volume 2024, pages 110–144.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.

Balachandra Devarangadi Sunil, Isheeta Sinha, Piyush Maheshwari, Shantanu Todmal, Shreyan Mallik, and Shuchi Mishra. 2026. Memory poisoning attack and defense on memory based llm-agents. *arXiv preprint arXiv:2601.05504*.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. 2024. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16022–16076.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Hanlin Wang, Chak Tou Leong, Jian Wang, and Wenjie Li. 2026. Seeing isn't believing: Mitigating belief inertia via active intervention in embodied agents. *arXiv preprint arXiv:2604.17252*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First conference on language modeling*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. 2025a. Agentracer: Who is inducing failure in the llm agentic systems? *arXiv preprint arXiv:2509.03312*.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and 1 others. 2025b. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. *arXiv preprint arXiv:2505.00212*.

Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, and 1 others. 2025. Where llm agents fail and how they can learn from failures. *arXiv preprint arXiv:2509.25370*.

Wei Zou, Mingwen Dong, Miguel Romero Calvo, Shuaichen Chang, Jiang Guo, Dongkyu Lee, Xing Niu, Xiaofei Ma, Yanjun Qi, and Jiarong Jiang. 2026. Poison once, exploit forever: Environment-injected memory poisoning attacks on web agents. *arXiv preprint arXiv:2604.02623*.

A Appendix

A.1 Phase 1 Detection Protocol

Detection Pipeline. Every trajectory passes through a two-step screening pipeline under the five-dimension rubric defined in Table 1 (main text). Step 1 applies a broad screen using a rubric-guided LLM evaluator to flag all candidate trajectories exhibiting narrative formation and narrative-organized behavior. Step 2 re-evaluates Step 1 positives under stricter criteria to suppress false positives and produces a conservative confirmed set. The Step 1 evaluator uses a permissive threshold (any dimension ≥ 1 triggers candidate flagging); the Step 2 evaluator requires at least two dimensions ≥ 2 and no dimension at 0 for confirmation. Damage level, confidence (high/medium/low), primary error type, solvability, and lock-in manifestation are recorded as auxiliary fields. Solvability filtering excludes cases where the task is fundamentally unsolvable from the start; confidence guides borderline adjudication for ambiguous trajectories.

Annotation Protocol. Each trajectory is independently scored by two LLM evaluators. Disagreements are resolved by a third evaluator with access to both initial scores and rationales. The five dimensions are scored on a 0–3 ordinal scale: 0 = absent, 1 = weak or ambiguous, 2 = moderate, 3 = strong or definitive. For CL-bench-Life, which lacks multi-turn action traces, NLI judgments rely primarily on Narrative Formation, State Adoption, and Counterevidence; the Behavioral Non-update dimension is scored only when the trajectory contains at least one substantive turn after the initial answer.

Prompts and Step Limits. The rubric is presented in the evaluator's system prompt together with dimension definitions, scoring anchors, and representative examples. Each evaluation call is limited to a single trajectory—the evaluator receives the full conversation history, tool calls, and environment observations. Step limits for agent benchmarks follow the default settings of each benchmark environment (AppWorld: 100 steps; ClawMark: as per task definition; WildClawBench: up to episode termination). Deduplication is applied at the $\langle$benchmark, task_id, model$\rangle$ level: multiple runs of the same configuration are treated as a single case, and the run with the median persistence score is retained for analysis in Phase 1; all runs are preserved for Phase 2 intervention experiments.

Inter-Annotator Agreement. Across the full Phase 1 annotation set, Step 2 inter-annotator agreement reaches Cohen's $\kappa = 0.73$ on the binary lock-in confirmation decision and a mean Spearman's $\rho = 0.68$ across the five rubric dimensions. Disagreement is concentrated in borderline cases with medium confidence, which are flagged for manual review.

A.2 Phase 2 Intervention Protocol

Case Selection. All Phase 2 cases are drawn from the Phase 1 NLI pool. We primarily sample from confirmed NLI cases and supplement with a small number of borderline cases and manually reviewed conservative NLI-like cases. Each case corresponds to a concrete $\langle$benchmark, task_id, model$\rangle$ failure instance. The intervention universe contains 735 cases: AppWorld 216, CL-bench-Life 288, ClawMark 133, WildClawBench 98.

Experimental Design. Each case serves as its own control. For every case, we start from the original trajectory up to the first lock-in evidence point identified in Phase 1, then inject one intervention. We run six intervention trajectories per case, crossing two trigger types and four correction levels, independently from the same pre-lock-in source state. Trigger conditions use meta-cognitive prompts without providing external corrective content; correction conditions inject an external correction with increasing specificity.

Two-Stage Post-Intervention Procedure. Intervention does not end at a single response. Each trajectory is evaluated in two stages. **Stage A** captures the model's immediate post-intervention response: revised answer, revised diagnosis, revised plan, or explicit refusal/non-update. **Stage B** requires the model to commit to one additional step immediately after Stage A. For behavior-oriented benchmarks (AppWorld, ClawMark, WildClawBench), this continuation step must specify a concrete next tool call or action intention with expected observation and goal. For CL-bench-Life, the continuation step must provide a revised answer and the next concrete step with justification. Stage B is included to distinguish verbal compliance from genuine trajectory update.

Intervention Prompts. Table 4 in the main text summarizes the six intervention arms. The full prompt text for each condition is as follows:

- **self_check**: "Review the trajectory above. Identify anything inappropriate, unsupported,

or inconsistent in your current approach. Do not assume your current interpretation is correct; explicitly point to what may be wrong."

- **meta_reflection**: "Pause and reflect: are you confident in your current approach? What evidence would disprove it? What assumptions are you relying on, and which of them may be unsupported by the task environment?"
- **L1 – Weak Recheck**: "Double-check your current approach. Return to the available evidence and verify each key assumption before continuing."
- **L2 – Claim Only**: "Your current core claim is likely wrong or inconsistent with the available evidence. Abandon your current assumption and re-evaluate the task."
- **L3 – Concrete Evidence**: "Your current narrative is contradicted by the following evidence: `{evidence}`. Re-evaluate the task based on this counterevidence and change your subsequent reasoning accordingly." The `{evidence}` placeholder is populated with specific contradictory evidence extracted from the task environment at the lock-in point.
- **L4 – Explicit Solution Path**: "Use the following corrected path: `{solution}`. Counterevidence: `{evidence}`. Abandon the original path and continue using the corrected path above. Your next step must follow this new path concretely."

Evaluation and Recovery Definition. We evaluate three layers: pre-intervention severity (persistence, outcome damage), post-intervention uptake (narrative change, action change, goal realignment, verification change, artifact change), and post-intervention outcome. A case is judged as *strictly recovered* only when all three hold: (1) the model abandons the original incorrect narrative and moves toward the correct direction; (2) the Stage B continuation shows a concrete updated next step or action path; (3) the judged task outcome improves. Cases that only acknowledge the intervention verbally, rewrite the answer without changing trajectory, or produce a repair plan without transitioning into a new step are treated as partial or surface responses.

Resource and Reproducibility. Code, evaluation prompts, and data processing scripts are available at https://github.com/Takamatsu-Hikaru/NLI_Code.

A.3 Discussion: Possible Underlying Mechanisms

We outline four candidate mechanisms that may contribute to Narrative Lock-in. These are offered as explanatory hypotheses rather than established causal accounts; rigorous attribution requires comparing lock-in incidence across different training paradigms, context management strategies, and state supervision mechanisms.

Autoregressive generation amplifies the prior effect of early claims. LLMs generate token by token, with each step conditioned on existing context. Once a model forms and writes down a high-confidence judgment early on, that judgment becomes an implicit constraint on subsequent interpretation, reasoning, and action planning. To maintain semantic coherence and local consistency, the model is more inclined to extend along the existing narrative than to actively reopen the problem space. Each subsequent turn that builds on the initial narrative further deepens the commitment, because the cost of unwinding grows with the number of decisions, sub-goals, and tool calls that have been anchored to it. In effect, autoregressive decoding converts a local misjudgment into a path-dependent trajectory from which the model cannot easily escape through local token-level corrections alone.

Training and post-training may reward coherent narratives over substantive revision. Pretraining corpora contain abundant complete, fluent, causally closed text structures, predisposing models to complete explanations and maintain contextual consistency. Post-training preference optimization often rewards clear, confident, actionable responses, which may reinforce the tendency to continue along an established path under uncertainty rather than explicitly suspending judgment, abandoning the current path, and reconstructing the diagnostic state. This creates an asymmetry: generating a confident continuation of a plausible-sounding but incorrect diagnosis looks more competent under standard preference criteria than producing a hesitant, self-correcting trajectory. The result is a behavioral disposition in which the agent would rather commit to a coherent story and execute it to completion than interrupt its own flow with epistemic second-guessing—precisely the pattern we observe across benchmarks in our experi-

ments (Sections 4–5).

Re-verification and task-state reconstruction carry higher generative cost. Overturning an early judgment requires isolating variables, reconstructing test conditions, distinguishing isomorphic from contaminated evidence, and reorganizing task objectives. By contrast, assimilating new evidence into the existing explanation is lower-cost and more consistent with the generative pressure for contextual continuity. The model may therefore substitute self-consistency for factual verification, treating "can be explained within the current context" as a proxy for "is more likely to be correct." This substitution is particularly dangerous in tool-augmented settings: each tool call produces raw output that could, in principle, falsify the current hypothesis, but interpreting that output requires the agent to actively entertain alternative framings. If the cost of re-evaluation exceeds the cost of assimilation, the agent will assimilate even when it should revise. Over many turns, this cost asymmetry compounds, making late-stage correction exponentially harder than early-stage verification.

LLMs are context-bound systems without a persistent epistemic state. Their "beliefs," "reflections," and "role perceptions" are primarily induced by the current context rather than maintained by a stable internal subject. When the context is already occupied by an early erroneous narrative, the model lacks a persistent verification mechanism independent of that context to flag that narrative lock-in may have occurred. Consequently, a local erroneous claim may continuously organize subsequent attention, evidence absorption, and action selection, producing systematic error persistence. This distinguishes Narrative Lock-in from simpler forms of hallucination or reasoning error: a hallucination is a single incorrect output, whereas lock-in is a *self-stabilizing regime* in which the context itself has been corrupted and the model's only window onto the task is through that corrupted lens. Without an external state representation that survives context overwrites, the agent has no ground from which to notice that its own trajectory has diverged from the task goal.

Human-in-the-loop remains the most effective intervention, and the hardest to scale. In practice, human oversight is currently one of the most reliable harness mechanisms for breaking Narrative Lock-in: a human operator can recognize when the agent is stuck in a story, inject clean counterevidence with explicit scope and source, and demand behavioral-level re-evaluation rather than textual acknowledgment. However, human intervention is costly, latency-prone, and does not scale to autonomous long-horizon deployments. A central motivation behind Nexus, and a key direction for future work, is to equip the Mentor with a meaningful subset of this oversight capability: detecting when the agent's trajectory has become narrative-constrained, issuing targeted interventions with verifiable behavioral criteria, and escalating to human attention only when automated intervention fails. In this sense, the Mentor is not intended to replace human judgment, but to absorb the routine, detectable class of lock-in failures that currently require manual correction, reserving human-in-the-loop for genuinely ambiguous or high-stakes cases.