BEYOND MEMORY: HARNESSING LONG-HORIZON AGENTS WITH EXPLICIT BELIEF STATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents can now undertake increasingly complex tasks, but the way they organize interaction history into memory does not ensure a coherent understanding of the current world. We introduce *PoS*, an inference-time framework that constructs and continually maintains explicit belief states as the agent's decision context. Each belief combines an estimate of the current world state with unresolved task requirements, making explicit what the agent still needs to learn and accomplish. To keep this belief reliable and actionable, *PoS* validates its consistency and monitors task progress to detect *Belief Trapping*, where the agent continues to act without making meaningful progress toward the goal. Recovery is then tailored to both the trapping pattern and the type of unresolved task requirement. Experiments on four benchmarks spanning execution and diagnosis show that *PoS* achieves the highest overall performance on every benchmark with all three LLM backbones. Ablations demonstrate the importance of consistency validation and recovery, while context-scaling experiments show resilience to context growth. Together, these results support belief construction and continual maintenance as a foundation for long-horizon context management beyond history retention and compression. Code is available at `https://anonymous.4open.science/r/PoS-321C`.

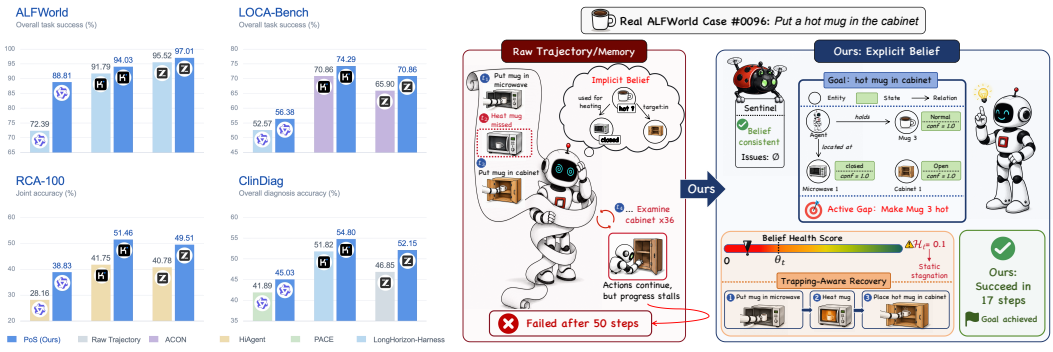


Figure 1: **Performance and motivation of *PoS*. Left:** *PoS* outperforms the strongest baseline for each backbone across four benchmarks. **Right:** a real ALFWorld case contrasts implicit belief reconstruction that leads to *Belief Trapping* with explicit belief maintenance in *PoS*, which enables consistency validation and recovery toward task completion.

1 INTRODUCTION

LLM agents are increasingly equipped with ultra-long context windows (Yang et al., 2025; Xu et al., 2026a; Hu et al., 2026), stronger reasoning and tool-use capabilities (Guo et al., 2025; Feng et al., 2026), external memory systems (Hu et al., 2025; Kang et al., 2025; Wei et al., 2026; Xu et al., 2026b), and sophisticated execution harnesses (Ma et al., 2026; Wan et al., 2026), substantially expanding the complexity and horizon of tasks they can undertake. However, longer interaction horizons expose a deeper challenge, as actions may alter the environment and new observations

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

may invalidate earlier inferences. Accumulated context thus mixes evidence about past states with intermediate judgments, making it increasingly difficult to identify which facts still hold, which inferences remain supported, and what remains unresolved. Raw trajectories preserve this evidence chronologically, while memory mechanisms select, compress, or reorganize it. While both are valuable, access to evidence alone does not ensure an explicit and coherent estimate of the current world that can be revised and checked across steps. Recent work observes that when task state remains implicit within a growing context, outdated or inconsistent information can persist across subsequent decisions (Ma et al., 2026). The next barrier to reliable long-horizon LLM agents is therefore not simply retaining more history, but maintaining an actionable representation of the current world.

A formal basis for this representation is the belief state $b_t(s) = P(s_t = s \mid h_t)$, the posterior over the current latent world state s_t given interaction history h_t. In a standard partially observable Markov decision process (POMDP), this belief is a sufficient statistic for decision-making and can be recursively updated after each action and observation (Kaelbling et al., 1998). Recent methods have begun to explicitly represent the belief state through goal-relative reflection, probabilistic hypotheses, structured graphs, or audited task records (Kim et al., 2025; Tang et al., 2026; Luo et al., 2026; Ma et al., 2026). Such representations make the agent's current understanding of the world available for inspection and revision. However, explicit belief construction is just the start, as an LLM-generated belief must satisfy two further requirements.

First, belief updates must remain internally consistent and supported by interaction evidence. Since these updates are generated by the LLM itself, they may introduce conflicting states or unsupported inferences that undermine subsequent decisions and ultimately lead to task failure. For instance, recording a microwave as both open and closed makes it unclear whether the robot should open it before placing a mug inside. Continual validation is thus essential to sustain the coherence and evidential support of the evolving belief state. Second, consistency alone does not prevent the failure mode we term *Belief Trapping*, where the agent continues to act without making meaningful progress toward the goal. For example, the agent may repeat ineffective actions, cycle through previously visited states, or gather information that does not help complete the task. Unproductive interactions will exhaust the available budget before achieving the goal, making timely detection and recovery necessary to restore meaningful task progress. T^3 detects belief deviation and truncates uninformative trajectories for reinforcement learning, but does not provide a recovery strategy within the ongoing episode (Zou et al., 2026). To our knowledge, no existing framework maintains a belief state with both continual consistency validation and online diagnosis and recovery from *Belief Trapping*.

We introduce **P**rogression **o**f **S**tates (*PoS*), a unified framework that integrates explicit belief modeling, consistency validation, and trapping-aware recovery for long-horizon agent decision-making. As illustrated in Figure 2, *PoS* continually updates a world state of task-relevant Entities, States, and Relations, along with epistemic and achievement gaps that specify what the agent still needs to learn or accomplish. The agent selects actions from the current belief, while a Belief Sentinel validates updates for internal consistency and support from interaction evidence. *PoS* also assesses whether belief evolution supports continued task progress using three measures: gap persistence, progress stagnation, and belief recurrence. It then performs factorized trapping diagnosis along two dimensions, agent dynamics and blocked gap type, and composes corresponding recovery constraints to recover from *Belief Trapping*.

We evaluate *PoS* using three LLM backbones on ALFWorld and LOCA-Bench for execution tasks and RCA-100 and ClinDiag for diagnosis tasks. *PoS* achieves the highest overall performance on all four benchmarks with each backbone, with relative gains over the strongest baseline reaching 22.68% on ALFWorld and 37.89% on RCA-100 joint accuracy. Ablation studies demonstrate the importance of consistency validation and trapping recovery beyond explicit belief construction. Further analyses examine trapping patterns and recovery effectiveness, robustness to context growth, and the tradeoff between token consumption and task performance.

2 Related Work

We organize prior approaches to long-horizon context management according to the decision context they expose to the agent: *memory as context* and *belief as context*.

2.1 MEMORY AS CONTEXT

ReAct (Yao et al., 2023) provides a foundational framework for LLM agents, conditioning each action on the growing raw interaction trajectory. Compression-based methods instead summarize accumulated context into compact forms: ReadAgent (Lee et al., 2024) constructs retrievable gist memories, ACON (Kang et al., 2025) optimizes guidelines for history compression, and SUPO (Lu et al., 2026) jointly learns periodic summarization and tool-use policies. Other methods impose structure on the retained context: HiAgent (Hu et al., 2025) organizes working memory around subgoals, COMPASS (Wan et al., 2026) maintains concise progress briefs across reasoning stages, and PACE (Wei et al., 2026) adjusts historical granularity according to its predicted relevance to the next action. Other approaches make context management itself an agent-controlled operation: MemGPT manages information across in-context and external memory through an operating-system-inspired hierarchy (Packer et al., 2023). CaT (Liu et al., 2026) exposes context compression as a callable tool over a structured workspace, AgentFold (Ye et al., 2026) performs proactive multi-scale folding of historical trajectories, and Sculptor (Li et al., 2026) provides explicit tools for actively curating the working context. Although these methods improve evidence accessibility and context efficiency, the resulting context remains a transformed record of historical evidence rather than a consistent estimate of the current world.

2.2 BELIEF AS CONTEXT

A parallel line of work studies how LLMs represent the current state of an evolving world (Gurnee & Tegmark, 2024). Mechanistic studies probe the world-state abstractions encoded by LLMs (Li et al., 2024) and show that state tracking of reasoning is itself a learned computation whose underlying mechanism affects systematic generalization (Li et al., 2025). Agent frameworks further externalize parts of this computation (Tang et al., 2024). QuBE (Kim et al., 2024) constructs a task-relevant belief through question answering; StateAct (Rozanov & Rei, 2025) maintains a chain of states throughout task execution; and ReflAct (Kim et al., 2025) grounds action selection in reflection over the current state relative to the goal. More structured approaches maintain particle beliefs over latent states and goals through hypothesis trees and Bayesian filtering (Tang et al., 2026), recursively update natural-language belief summaries (Lidayan et al., 2025), or combine causal graphs with state machines as explicit belief (Luo et al., 2026). LongHorizon-Harness (Ma et al., 2026) externalizes task state as audited records of requirements, artifacts, and facts, preserving verified progress through a Manage-Execute-Audit loop. However, it relies on independently verifiable intermediate results and is therefore less suitable when meaningful progress consists of uncertain inferences rather than observable outcomes. Beyond belief construction, recent work has also begun to address failure modes related to *Belief Trapping*. T^3 (Zou et al., 2026) detects excessive belief deviation and truncates uninformative trajectories during reinforcement learning, while CGDP (Kausik et al., 2026) uses an exhaustion gate to terminate unproductive context gathering. These approaches address stalled progress through truncation or termination, without attempting to restore progress during inference. In contrast, ***PoS*** maintains and validates a task-conditioned belief and uses factorized recovery to redirect subsequent actions.

3 METHODOLOGY

PoS comprises three components: Belief Modeling constructs a task-conditioned belief (Section 3.1), Belief-Guided Interaction selects actions and validates belief updates (Section 3.2), and Trapping-Aware Recovery detects and recovers from *Belief Trapping* (Section 3.3).

3.1 BELIEF MODELING

The interaction between a long-horizon agent and its environment can be formulated as a POMDP. At step t, the environment has a latent world state s_t that is not directly observable. Given the interaction history h_t, the belief state $b_t(s)$ represents the posterior distribution over s_t and provides the basis for selecting an action a_t. Following this formulation, ***PoS*** maintains a structured, task-conditioned belief B_t as the agent's explicit decision context:

$$B_t = \left(W_t, G, \Delta_t^E, \Delta_t^A\right), \tag{1}$$

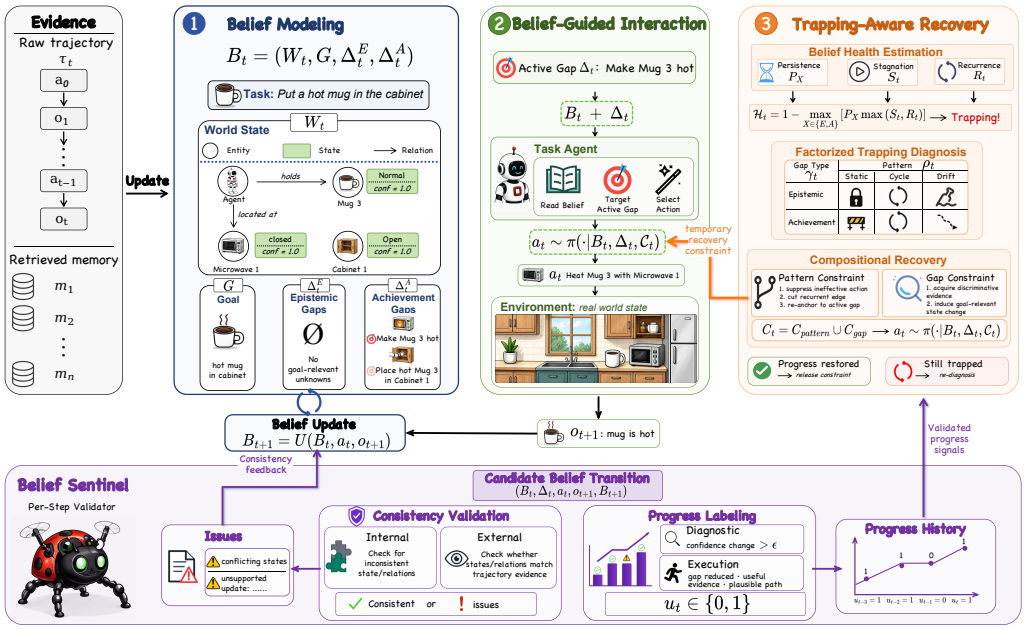


Figure 2: **Overview of *PoS*.** Belief Modeling constructs and incrementally updates a task-conditioned Belief from interaction evidence (Section 3.1). Belief-Guided Interaction conditions each action on the current Belief and an Active Gap, while the Belief Sentinel validates consistency and records task-relevant progress (Section 3.2). Trapping-Aware Recovery estimates Belief health, diagnoses heterogeneous trapping conditions, and composes recovery constraints to restore progress (Section 3.3).

where W_t is a structured estimate of the task-relevant world state, G specifies the task objective, and Δ_t^E and Δ_t^A capture unresolved epistemic and achievement gaps, respectively.

World state (W_t). Prior work often represents an environment through entities and their relations (Anokhin et al., 2025; Gu et al., 2024). For long-horizon decision making, however, entity identity and connectivity alone are insufficient: an entity's condition may change over time, directly affecting which actions are feasible and whether the task goal is satisfied. *PoS* therefore represents W_t with an *Entity–State–Relation* structure that applies across domains and makes such changes explicit. Within this structure, an *Entity* denotes an object or agent involved in the task, while a *State* describes the agent's current understanding of that Entity in natural language. A directed *Relation* captures how Entities and States are connected, including relations between two Entities, two States, or an Entity and a State. Each State and Relation is annotated with its provenance and confidence, together with supporting evidence when available. The natural-language descriptions in W_t provide the flexibility to express domain-specific semantics, while the explicit structure supports consistent belief updates and validation. Rather than attempting to model the environment exhaustively, W_t retains only records that can affect action feasibility, outcome assessment, or satisfaction of G.

Goal-conditioned components (G, Δ_t^E, and Δ_t^A). Although W_t captures the task-relevant world state, it does not by itself indicate what remains unresolved for completing the task. *PoS* evaluates W_t against the task objective and completion criteria specified by G and represents the unresolved requirements as two complementary gap sets. Epistemic gaps Δ_t^E capture information that remains unknown or insufficiently supported, whereas achievement gaps Δ_t^A capture discrepancies between the current belief and the desired goal state that must be resolved through action. This distinction separates what the agent still needs to learn from what it still needs to accomplish. Together, W_t, G, Δ_t^E, and Δ_t^A allow B_t to represent both the agent's current understanding of the world and the remaining requirements for goal completion, making it an actionable decision state.

3.2 BELIEF-GUIDED INTERACTION

Although B_t represents both the current world and the unresolved task requirements, it may contain multiple epistemic and achievement gaps. Giving equal priority to all gaps in $\Delta_t^E \cup \Delta_t^A$ leaves the agent without a clear focus and may cause it to shift between gaps without making sustained progress on any of them. ***PoS*** therefore retains B_t as the complete decision state while selecting an active gap $\Delta_t \in \Delta_t^E \cup \Delta_t^A$ that are the most valuable to address given the current world state and task objective. The active gap establishes the current focus without discarding the broader world understanding and task context represented by B_t. We then formulate action selection as:

$$a_t \sim \pi\left(\cdot \mid B_t, \Delta_t, \mathcal{C}_t\right). \tag{2}$$

Here, $\mathcal{C}_t$ denotes the recovery constraint introduced in Section 3.3. It is empty during normal interaction and is applied only when ***Belief Trapping*** is detected, as an additional constraint on action selection. After executing a_t, the task agent receives the resulting observation o_{t+1} from the environment.

The new observation o_{t+1} provides evidence to update the belief. The task agent incorporates this evidence into B_t to form an unverified candidate $\widetilde{B}_{t+1}$, which cannot yet be committed as B_{t+1}. Because $\widetilde{B}_{t+1}$ is produced by the task agent itself, it may inherit the agent's erroneous assumptions, reasoning errors, or hallucinations. Such errors can produce two forms of inconsistency: internal and external. Internal inconsistency arises when $\widetilde{B}_{t+1}$ contains mutually conflicting content, such as incompatible States assigned to the same Entity. External inconsistency arises when $\widetilde{B}_{t+1}$ contradicts the latest observation or other available interaction evidence. ***PoS*** therefore introduces the ***Belief Sentinel*** to validate $\widetilde{B}_{t+1}$. It audits newly added or modified Entities, States, Relations, and Gaps, and returns potential consistency issues to the task agent. The task agent evaluates this feedback against the available interaction evidence and revises $\widetilde{B}_{t+1}$ before committing the resulting belief as B_{t+1}. The complete belief update is expressed as

$$\widetilde{B}_{t+1} = \mathcal{U}\left(B_t, a_t, o_{t+1}\right) \xrightarrow{\textit{Belief Sentinel}} B_{t+1}, \tag{3}$$

where $\mathcal{U}$ denotes the candidate belief update performed by the task agent, and the arrow represents Belief Sentinel validation followed by evidence-based revision by the task agent. The resulting B_{t+1} serves as the decision state for the next step and the basis for belief health estimation in Section 3.3, helping reduce the risk that erroneous belief updates are mistaken for actual interaction dynamics.

3.3 TRAPPING-AWARE RECOVERY

To assess task progress, ***PoS*** tracks changes in the world state and unresolved requirements across successive validated belief transitions. When ***Belief Trapping*** is detected, it performs factorized diagnosis of the trapping dynamics and blocked gap type, then composes corresponding recovery constraints to guide subsequent action selection.

Belief health estimation ($\mathcal{H}_t$). ***PoS*** estimates belief health at two temporal scales. At the step level, each validated transition $B_{t-1} \to B_t$ is assigned a progress label $u_t \in \{0, 1\}$. For diagnostic tasks, let $c_t(x)$ denote the confidence distribution over inferred States and Relations. We measure progress by aggregating absolute confidence change between consecutive beliefs:

$$d_t^{\text{diag}} = \frac{1}{2} \sum_x \left| c_t(x) - c_{t-1}(x) \right|, \qquad u_t = \mathbb{I}\left[d_t^{\text{diag}} > \epsilon \right], \tag{4}$$

where ϵ is a predefined threshold that filters out minor confidence fluctuations. For execution tasks, the ***Belief Sentinel*** assigns $u_t = 1$ when the transition reduces the active gap Δ_t, acquires information needed to resolve it, or advances a plausible path toward the goal; otherwise, $u_t = 0$.

However, a single transition is insufficient to determine whether the agent has become trapped. ***PoS*** therefore aggregates belief dynamics over the K most recent validated transitions, $\mathcal{W}_t = \{t - K + 1, \ldots, t\}$. Within this window, ***PoS*** measures three complementary signals. ***(1) Gap Persistence*** P_X, where $X \in \{E, A\}$, measures the fraction of initially unresolved gaps of type X that remain unresolved throughout the window. ***(2) Progress Stagnation*** S_t measures the fraction of transitions with no recorded task progress. ***(3) Belief Recurrence*** R_t measures how frequently

the world state relevant to the active gap repeats, taking the highest recurrence rate across candidate temporal lags. Let W_τ^Δ denote the projection of W_τ onto the current active gap Δ_t, and let d_J denote the Jaccard distance. We measure world state change using d_W, the average Jaccard distance between the corresponding Entity, State, and Relation sets.

$$P_X = \frac{\left|\bigcap_{\tau=t-K+1}^{t} \Delta_\tau^X\right|}{\left|\Delta_{t-K+1}^X\right|}, \qquad S_t = 1 - \frac{1}{K}\sum_{\tau=t-K+1}^{t} u_\tau, \tag{5a}$$

$$d_W\left(W_i^\Delta, W_j^\Delta\right) = \frac{1}{3}\sum_{\mathcal{Y}} d_J\left(\mathcal{Y}_i^\Delta, \mathcal{Y}_j^\Delta\right), \quad R_t = \max_{1\le\ell\le L} \frac{1}{K-\ell} \sum_{\tau=t-K+1+\ell}^{t} \mathbb{I}\left[d_W\left(W_\tau^\Delta, W_{\tau-\ell}^\Delta\right) \le \epsilon_W\right], \tag{5b}$$

$$\mathcal{H}_t = 1 - \max_{X\in\{E,A\}} \left[P_X \max\left(S_t, R_t\right)\right]. \tag{5c}$$

We set $P_X = 0$ when $\Delta_{t-K+1}^X = \varnothing$. Here, $\mathcal{Y} \in \{\mathcal{E}, \mathcal{S}, \mathcal{R}\}$ indexes the aligned and semantically matched Entity, State, and Relation sets, while L and ϵ_W denote the maximum lag and recurrence threshold, respectively. The resulting health score captures whether a persistent unresolved gap is accompanied by progress stagnation or belief recurrence. (Detailed explanations in Appendix B.3) Accordingly, *PoS* detects *Belief Trapping* when $\mathcal{H}_t \le \theta_H$.

Factorized trapping diagnosis. Once *Belief Trapping* is detected, *PoS* examines validated belief transitions within the same window $\mathcal{W}_t$ to diagnose it along two complementary dimensions: the agent dynamics pattern and the blocked gap type. First, it classifies the agent dynamics as:

$$\rho_t = \begin{cases} \text{Static}, & d_W(W_t, W_{t-1}) \le \epsilon_W, \\ \text{Cycle}, & R_t \ge \theta_R \ \wedge \ k_t^* > 1, \\ \text{Drift}, & d_W\left(W_t^\Delta, W_{t-1}^\Delta\right) \le \epsilon_W, \end{cases} \tag{6}$$

where the conditions are evaluated in order, k_t^* denotes the dominant recurrence lag underlying R_t, and θ_R is the recurrence threshold for identifying periodic behavior. Under this hierarchy, Static indicates that the world state remains locally unchanged. Among non-static cases, Cycle captures periodic recurrence, whereas Drift captures changes in the world state that do not advance its active-gap projection. Second, *PoS* identifies the blocked gap type by comparing P_E and P_A among nonempty gap sets. The type with higher persistence is selected: $X = E$ indicates blocked epistemic progress, whereas $X = A$ indicates blocked achievement progress. Together, the agent dynamics pattern specifies how to escape the current trapping state, while the blocked gap type specifies what progress recovery should restore.

Recovery constraints. Based on the diagnosed dynamics pattern and blocked gap type, *PoS* combines a pattern-specific escape constraint with a gap-specific progress constraint:

$$\mathcal{C}_t = \mathcal{C}_{\text{pattern}}\left(\rho_t, \mathcal{W}_t\right) \cup \mathcal{C}_{\text{gap}}\left(X, B_t, \Delta_t\right). \tag{7}$$

For Static, the pattern constraint suppresses the ineffective state–action transition. For Cycle, it breaks a recurrent transition identified by k_t^*. For Drift, it re-anchors action selection to the active gap. The gap-specific constraint instead requires new discriminative evidence when $X = E$ and a task-relevant state change when $X = A$. During recovery, the active gap Δ_t remains unchanged, while $\mathcal{C}_t$ is applied as an additional constraint in Eq. 2. After the resulting transition is validated, *PoS* recomputes $\mathcal{H}_t$. The recovery constraints are released when $\mathcal{H}_t > \theta_H$; otherwise, *PoS* updates the diagnosis and corresponding constraints.

4 Experiments

Benchmarks. We evaluate *PoS* on four long-horizon benchmarks covering two complementary forms of agentic decision making. ALFWorld (Shridhar et al., 2021) and LOCA-Bench (Zeng et al., 2026) are *goal-directed task-execution* benchmarks, in which the agent repeatedly acts in the environment to bring the current world into a goal-satisfying state. RCA-100 (Cai et al., 2026) and ClinDiag (Chen et al., 2026) are *evidence-seeking diagnosis* benchmarks, where the agent interacts with the environment to acquire discriminative evidence, update competing hypotheses, and ultimately identify the most plausible diagnosis.

Table 1: Main results across four benchmarks. We report overall task success for ALFWorld (ALF) and LOCA-Bench (LOCA), joint accuracy for RCA-100 (RCA), and overall diagnosis accuracy for ClinDiag (Clin), all in %. Best performance is bold. Fine-grained results in Table 9

Method	Qwen3.7-Plus				Kimi-K3				GLM-5.3			
	ALF	LOCA	RCA	Clin	ALF	LOCA	RCA	Clin	ALF	LOCA	RCA	Clin
Raw Trajectory	62.69	43.62	24.27	38.91	81.34	69.71	36.89	49.17	93.28	62.86	32.04	46.85
ACON	66.42	49.90	27.18	39.74	81.34	70.86	40.78	38.58	94.78	65.90	33.98	46.19
PACE	67.91	17.33	28.16	41.89	84.33	41.52	35.92	50.17	94.78	38.29	37.86	43.38
HiAgent	66.42	24.57	28.16	40.07	88.81	49.52	41.75	48.51	94.78	45.90	40.78	44.87
LongHorizon-Harness	72.39	52.57	26.21	40.56	91.79	67.62	38.83	51.82	95.52	64.38	31.07	45.53
PoS (OURS)	**88.81**	**56.38**	**38.83**	**45.03**	**94.03**	**74.29**	**51.46**	**54.80**	**97.01**	**70.86**	**49.51**	**52.15**
- Consistency Validation	73.88	44.57	31.07	44.54	91.79	67.43	48.54	54.47	94.78	62.67	47.57	51.49
- Trapping Diagnosis	73.13	48.19	33.98	42.38	88.81	69.52	46.60	50.83	92.54	65.71	42.72	48.51

Baselines. We adopt ReAct (Yao et al., 2023) as the common harness for context-management baselines, using its simple reasoning–action loop to keep the interaction protocol consistent while varying the context supplied for action selection. *Raw Trajectory* directly provides the complete, chronologically ordered action–observation history. ACON (Kang et al., 2025) and PACE (Wei et al., 2026) manage the growing context through compression and adaptive retention, respectively. HiAgent (Hu et al., 2025) organizes completed interactions around subgoals and maintains a hierarchical working memory. LongHorizon-Harness (Ma et al., 2026) maintains an audited task state and coordinates long-horizon execution through a Manage–Execute–Audit architecture. Together, these baselines cover raw history, compressed and adaptively retained context, hierarchical working memory, and explicitly maintained task state. LongHorizon-Harness retains its native control architecture, while the remaining baselines operate through the common ReAct harness.

Implementation details. We evaluate all methods with three backbone models: `Qwen3.7-Plus`, `Kimi-K3`, and `GLM-5.3`. Within each backbone setting, all LLM components of *PoS* and all LLM-based components of the baselines use the same evaluated backbone. All LLM calls use deterministic decoding with temperature 0. Complete benchmark protocols and budgets, baseline implementations, and method-specific hyperparameters are provided in the appendix D.

4.1 Main Results

(1) Existing context management does not consistently improve upon raw trajectories. Table 1 reveals substantial variations in baseline performance across tasks and backbones. PACE remains competitive on ALFWorld and RCA-100, but falls 24.57–28.19 points below Raw Trajectory on LOCA-Bench. LongHorizon-Harness performs well on LOCA-Bench, where explicit requirements and generated artifacts support intermediate verification, but delivers less consistent gains on RCA-100 and ClinDiag, as progress depends on refining uncertain diagnostic hypotheses rather than producing directly verifiable artifacts. On ClinDiag with `GLM-5.3`, none of the context-management baselines outperforms Raw Trajectory. These reversals suggest that additional compression and organization do not consistently benefit modern LLMs with strong long-context capabilities, as information loss may offset the benefits of reducing redundancy.

(2) *PoS* consistently outperforms existing methods across diverse tasks and backbones. In contrast, *PoS* achieves the highest overall performance on all four benchmarks under all three backbones. Compared with the strongest baseline with the same backbone, *PoS* achieves relative gains of up to 22.68% on ALFWorld, 7.53% on LOCA-Bench, 37.89% on RCA-100 joint accuracy, and 11.31% on ClinDiag. The gains span both execution and diagnosis tasks, indicating that they are not tied to a particular task structure. These results highlight the importance of maintaining an accurate, task-conditioned belief about the current world state and unresolved requirements, beyond retaining and organizing interaction history.

(3) Effective belief modeling requires maintaining both consistency and health. Removing Consistency Validation lowers performance across all four benchmarks to varying degrees, with losses reaching 14.93 points on ALFWorld and 11.81 points on LOCA-Bench, compared with 0.33–0.66 points on ClinDiag. This pattern is consistent with the greater risk of inconsistent belief updates when complex interactions require the agent to reconcile changing states and observations across steps. Removing Trapping Diagnosis and its associated recovery also degrades performance across all four benchmarks, including losses of 4.85–6.79 points on RCA-100 and 2.65–3.97 points on ClinDiag. Recovery complements validation by redirecting actions toward unresolved requirements when continued interaction no longer advances the task. Together, these results demonstrate the value of maintaining both belief consistency and task progress, beyond constructing an explicit representation.

4.2 Belief Trapping Analysis

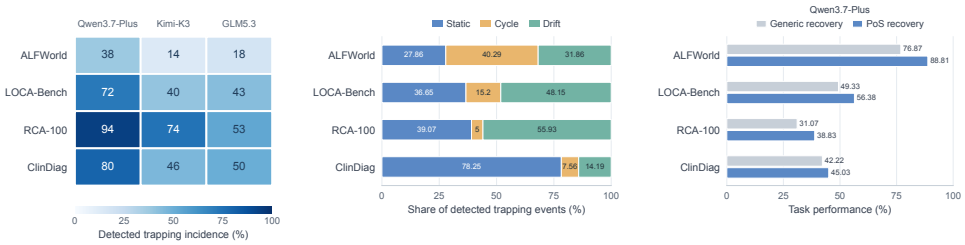


Figure 3: **Belief Trapping and recovery across benchmarks. Left:** Percentage of episodes with detected trapping under each backbone. **Middle:** Distribution of trapping patterns by benchmark. **Right:** Final task performance over all evaluated cases with Qwen3.7-Plus, comparing *PoS*'s factorized recovery with a generic recovery prompt.

Figure 3 summarizes detected trapping incidence (left), the distribution of trapping patterns (middle), and the effect of recovery strategies after trapping is detected (right). It reveals that trapping is common across different settings, although stronger backbones generally exhibit lower incidence. However, lower trapping incidence does not necessarily imply higher final accuracy: on RCA-100, `Kimi-K3` exhibits more trapping than `GLM-5.3` (74% vs. 53%) but achieves higher Joint accuracy (Table 1). The dominant trapping pattern also varies across benchmarks, with cycles most frequent in ALFWorld, drift in LOCA-Bench and RCA-100 (55.93%), and static stagnation in ClinDiag (78.25%). These differences reflect how unproductive interactions affect belief transition. Embodied actions may revisit prior states, multisource evidence may update beliefs without advancing the active gap, and non-discriminative clinical investigations may leave diagnostic beliefs unchanged. To assess the value of factorized recovery, we compare *PoS* with a variant that replaces factorized recovery constraints with a generic prompt asking the agent to recover. Figure 3 (right) shows that *PoS*'s recovery strategy outperforms this variant across all four benchmarks. These results demonstrate that effective recovery goes beyond interrupting unproductive behavior, with factorized recovery tailored to the trapping dynamics and blocked gap type.

4.3 Cost–Performance Tradeoff

Table 2 examines the cost–performance tradeoff on RCA-100 with `Qwen3.7-Plus`. Compared with Raw Trajectory, *PoS* improves joint accuracy from 24.27% to 38.83%, while reducing Task Agent token consumption by 20.9%. However, total consumption increases to 5.06×, as additional computation is devoted to belief construction and maintenance. Removing Consistency Validation substantially reduces Sentinel overhead and saves 35.6% of total tokens, but sacrifices 7.76 points in joint accuracy. By contrast, removing Trapping Diagnosis and its associated recovery saves only 9.2% of total tokens, while increasing Task Agent consumption and reducing joint accuracy by 4.85 points. Both ablations therefore reduce overall cost but leave the Task Agent consuming more tokens with less accurate final diagnoses. These results highlight the value of investing additional computation in belief maintenance to improve subsequent decisions, rather than minimizing token consumption alone.

Table 2: Cost–performance tradeoff on RCA-100. We report joint accuracy (%) and mean token consumption per episode in thousands (K), averaged over all 103 cases. Token counts include input and output tokens for each component.

Backbone	Method	Joint (%)	Tokens per episode (K)			
			Task Agent	Belief	Sentinel	**Total**
Qwen3.7-Plus	Raw Trajectory	24.27	355.70	0.00	0.00	355.70
	PoS (Ours)	**38.83**	281.20	696.69	822.24	1800.13
	- Consistency Validation	31.07	299.82	733.41	125.30	1158.53
	- Trapping Diagnosis	33.98	327.40	657.47	649.11	1633.98

4.4 Scaling under Context Growth

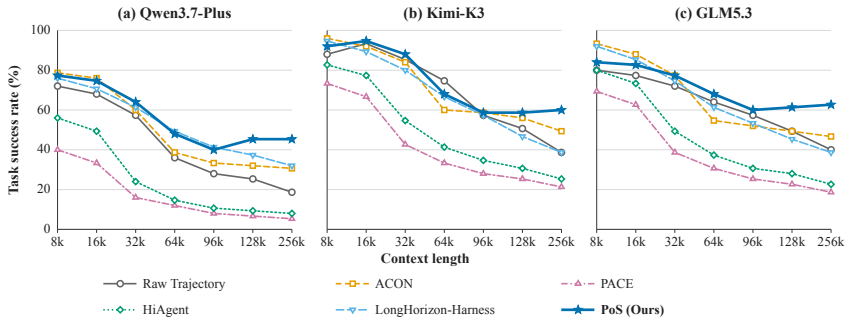


Figure 4: **Success rates on LOCA-Bench as context grows from 8K to 256K across backbones.**

Figure 4 compares task success on LOCA-Bench as environment descriptions grow from 8K to 256K across three backbones. Although performance generally declines with context length, ACON shows a more gradual decline than Raw Trajectory at longer contexts. By compressing accumulated interaction history whenever the context exceeds a predefined threshold, ACON reduces redundancy, whose increasing burden can make compression beneficial despite potential information loss. Meanwhile, *PoS* maintains broadly stable performance from 96K to 256K, exceeding the strongest baseline by 10.67–16.00 points at 256K. By integrating new observations into an existing world state and unresolved requirements, *PoS* grounds each decision in the current task conditions rather than requiring the agent to reconstruct them from an increasingly long interaction history. This incremental maintenance reduces repeated interpretation of dispersed or outdated observations, while consistency validation and progress monitoring help prevent unsupported updates and stalled interactions from persisting. Together, these results suggest that effective context management should balance reducing historical redundancy with maintaining a reliable and actionable understanding of the current world.

5 Conclusion

In this work, we showed that explicit belief construction alone is insufficient for reliable long-horizon decision-making. We proposed *PoS*, a framework that constructs task-conditioned belief states and continually maintains them through consistency validation and factorized recovery from *Belief Trapping*. Experiments across four benchmarks and three LLM backbones demonstrate consistent improvements in execution and diagnosis, together with resilience to context growth. Although *PoS* uses more tokens, the resulting gains demonstrate the potential of investing inference-time computation in belief maintenance. Reducing this overhead while preserving the gains in agent performance is an important direction for future work.

6 AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing our methods and editing the manuscript to improve its grammar, clarity, and readability. We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims or assist with their proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, clean or reformat datasets, conduct qualitative or thematic data analysis, or interpret results. We have reviewed all AI-assisted work. We checked the AI-assisted implementations for correctness and consistency with the methods described in the paper. We also checked the translations and edits against the original text to ensure that they preserve the intended technical meaning and do not introduce unsupported claims or alter citations. We take responsibility for the final content of this work, including all text and code produced with the assistance of generative AI.

REFERENCES

Petr Anokhin, Nikita Semenov, Artyom Y. Sorokin, Dmitry Evseev, Andrey Kravchenko, Mikhail Burtsev, and Evgeny Burnaev. Arigraph: Learning knowledge graph world models with episodic memory for LLM agents. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pp. 12–20. ijcai.org, 2025. doi: 10.24963/IJCAI.2025/2. URL https://doi.org/10.24963/ijcai.2025/2.

Yuanhong Cai, Xiaohui Nie, Kanglin Yin, Changhua Pei, Yongqian Sun, Shenglin Zhang, Haibin Liu, Guiyang Liu, Xidao Wen, Fang Situ, et al. A multi-dataset benchmark for evaluating llm agents in microservice failure diagnosis. *arXiv preprint arXiv:2606.29193*, 2026.

Xi Chen, Hanyu Zhou, Huahui Yi, Mingke You, WeiZhi Liu, Li Wang, Ziyuan Qin, Hairui Li, Xue Zhang, Yingman Guo, et al. Grounding large language models in clinical diagnostics. *Nature Communications*, 2026.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. In *International Conference on Learning Representations*, volume 2026, pp. 37909–37926, 2026.

Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pp. 5021–5028. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10610243. URL https://doi.org/10.1109/ICRA57147.2024.10610243.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng

Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, Sept 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *International Conference on Learning Representations*, volume 2024, pp. 2483–2503, 2024.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32779–32798, 2025.

Xiang Hu, Zhanchao Zhou, Ruiqi Liang, Zehuan Li, Wei Wu, and Jianguo Li. Every token counts: Generalizing 16M ultra-long context in large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10208–10220, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.464. URL https://aclanthology.org/2026.acl-long.464/.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin A Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. Acon: Optimizing context compression for long-horizon llm agents. *arXiv preprint arXiv:2510.00615*, 2025.

Chinmaya Kausik, Adith Swaminathan, and Nathan Kallus. The context gathering decision process: A pomdp framework for agentic search. *arXiv preprint arXiv:2605.07042*, 2026.

Jeonghye Kim, Sojeong Rhee, Minbeom Kim, Dohyung Kim, Sangmook Lee, Youngchul Sung, and Kyomin Jung. Reflact: World-grounded decision making in llm agents via goal-state reflection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 33421–33453, 2025.

Minsoo Kim, Jongyoon Kim, Jihyuk Kim, and Seung-won Hwang. Qube: Question-based belief enhancement for agentic llm reasoning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 21403–21423, 2024.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John F. Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 26396–26415. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/lee24c.html.

Belinda Z. Li, Zifan Carl Guo, and Jacob Andreas. (How) do language models track state? In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34429–34452. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/li25r.html.

Mo Li, LH Xu, Qitai Tan, Long Ma, Hongyong Song, Ting Cao, and Yunxin Liu. Sculptor: Empowering llms with cognitive agency via active context management. In *International Conference on Learning Representations*, volume 2026, pp. 153411–153440, 2026.

Zichao Li, Yanshuai Cao, and Jackie C Cheung. Do llms build world representations? probing through the lens of state abstraction. *Advances in Neural Information Processing Systems*, 37: 98009–98032, 2024.

Aly Lidayan, Jakob Bjorner, Satvik Golechha, Kartik Goyal, and Alane Suhr. Abbel: Learning natural-language belief states for memory-efficient interaction. *arXiv preprint arXiv:2512.20111*, 2025.

Shukai Liu, Bo Jiang, Jian Yang, Yizhi LI, Jinyang Guo, Xianglong Liu, and Bryan Dai. Context as a tool: Context management for long-horizon SWE-agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 20604–20617, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1032. URL https://aclanthology.org/2026.findings-acl.1032/.

Miao Lu, Weiwei Sun, Weihua Du, Zhan Ling, Xuesong Yao, Kang Liu, and Jiecao Chen. Beyond the context window: Scaling agentic RL via end-to-end optimized context compression. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21074–21125, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.966. URL https://aclanthology.org/2026.acl-long.966/.

Yu Luo, Rongchen Gao, Lu Teng, Xidao Wen, Jiamin Jiang, Qingliang Zhang, Yongqian Sun, Shenglin Zhang, Jiasong Feng, Tong Liu, et al. Graph of states: Solving abductive tasks with large language models. *arXiv preprint arXiv:2603.21250*, 2026.

Ziyu Ma, Hailang Huang, Shun Zou, Yong Wang, Shidong Yang, Yiming Hu, Fei Wei, and Xiangxiang Chu. Longhorizon-harness: Advancing long-horizon agents for real-world tasks, 2026.

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

Nikolai Rozanov and Marek Rei. Stateact: Enhancing llm base agents via self-prompting and state-tracking. In *Proceedings of the 1st Workshop for Research on Agent Language Models (REALM 2025)*, pp. 367–385, 2025.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. {ALFW}orld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0IOX0YcCdTn.

Hao Tang, Darren Key, and Kevin Ellis. Worldcoder, a model-based llm agent: Building world models by writing code and interacting with the environment. *Advances in Neural Information Processing Systems*, 37:70148–70212, 2024.

Wenjing Tang, Xinyu He, Yongxi Huang, Yunxiao Xiao, Cewu Lu, and Panpan Cai. Tru-pomdp: Task planning under uncertainty via tree of hypotheses and open-ended pomdps. *Advances in Neural Information Processing Systems*, 38:55367–55396, 2026.

Guangya Wan, Mingyang Ling, Xiaoqi Ren, Rujun Han, Sheng Li, and Zizhao Zhang. COMPASS: Enhancing agent long-horizon reasoning with evolving context. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3360–3380, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.152. URL https://aclanthology.org/2026.acl-long.152/.

Lei Wei, Xiao Peng, Guannan Zhang, Chenhao Jiang, Hongyu Li, Lanbo Lin, Yuanwu Xu, Jiayao Liu, Kesu Wang, Bin Wang, et al. Pace: Predictive adaptive context extraction for long-horizon llm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27184–27199, 2026.

Chejian Xu, Wei Ping, Peng Xu, Zihan Liu, Boxin Wang, Mohammad Shoeybi, Bo Li, and Bryan Catanzaro. From 128K to 4M: Efficient training of ultra-long context large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 13122–13133, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.640. URL https://aclanthology.org/2026.findings-acl.640/.

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

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, Weijia Xu, Wenbiao Yin, Wenyuan Yu, Xiafei Qiu, Xingzhang Ren, Xinlong Yang, Yong Li, Zhiying Xu, and Zipeng Zhang. Qwen2.5-1m technical report, 2025. URL https://arxiv.org/abs/2501.15383.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, et al. Agentfold: Long-horizon web agents with proactive context folding. In *The Fourteenth International Conference on Learning Representations*, 2026.

Weihao Zeng, Yuzhen Huang, and Junxian He. Loca-bench: Benchmarking language agents under controllable and extreme context growth. *arXiv preprint arXiv:2602.07962*, 2026.

Deyu Zou, Yongqiang Chen, Jianxiang Wang, Garry Yang, Mufei Li, Qing Da, James Cheng, Pan Li, and Yu Gong. Reducing belief deviation in reinforcement learning for active reasoning of llm agents. In *International Conference on Learning Representations*, volume 2026, pp. 21511–21545, 2026.

CONTENTS

1 Introduction **1**

2 Related Work **2**

2.1 Memory as Context 3

2.2 Belief as Context 3

3 Methodology **3**

3.1 Belief Modeling 3

3.2 Belief-Guided Interaction 5

3.3 Trapping-Aware Recovery 5

4 Experiments **6**

4.1 Main Results 7

4.2 Belief Trapping Analysis 8

4.3 Cost–Performance Tradeoff 8

4.4 Scaling under Context Growth 9

5 Conclusion **9**

6 AI Use Statement **10**

References **10**

A Notation Summary **16**

B Supplementary Method Details **17**

B.1 End-to-End Workflow of *PoS* 17

B.2 Belief Trapping Detection and Recovery 17

B.3 Design Rationale for Belief Trapping 18

C Benchmark Details and Evaluation Protocols **19**

C.1 Task Taxonomy 19

C.2 Dataset Statistics and Splits 19

C.3 Interactive Evaluation Protocols 20

C.4 Metrics, Budgets, and Termination 21

D Implementation and Reproducibility Details **22**

D.1 Backbone and Agent Configuration 22

D.2 Baseline Selection 22

D.3 Baseline Implementation 22

D.4 PoS Hyperparameters 24

E Fine-Grained Main Results **24**

F Case Study **26**

G Limitations and Future Directions **27**

A NOTATION SUMMARY

Table 3 summarizes the notation used throughout the methodology and supplementary material.

Table 3: Summary of the principal notation used in *PoS*.

Symbol	Meaning	Domain / Notes
Environment and interaction		
s_t	Latent world state at step t	Not directly observable by the agent.
h_t	Interaction history at step t	Contains previous actions and observations.
$a_t,\ o_{t+1}$	Agent action and resulting observation	The environment returns o_{t+1} after action a_t.
T	Maximum interaction budget	Defined separately for each benchmark.
Belief representation and update		
$b_t(s)$	Posterior belief over the latent world state	$b_t(s) = P(s_t = s \mid h_t)$ in the POMDP formulation.
B_t	Task-conditioned belief state	$B_t = (W_t, G, \Delta_t^E, \Delta_t^A)$.
W_t	Task-relevant world state	Structured through Entities, States, and Relations.
G	Task goal	Specifies the objective and completion criteria.
$\Delta_t^E,\ \Delta_t^A$	Epistemic and achievement gap sets	Information to clarify and goal conditions to achieve.
Δ_t	Active gap at step t	$\Delta_t \in \Delta_t^E \cup \Delta_t^A$; retained until resolved.
$\widetilde{B}_{t+1}$	Candidate belief update	Validated before being committed as B_{t+1}.
$\mathcal{U}$	Candidate belief-update operator	$\widetilde{B}_{t+1} = \mathcal{U}(B_t, a_t, o_{t+1})$.
π	Action-selection policy	Conditions on B_t, Δ_t, and $\mathcal{C}_t$.
Progress and belief health		
$c_t(x)$	Confidence assigned to inferred content x	Independent confidence scores for goal-relevant inferred States and Relations, without normalization across records.
d_t^{diag}	Diagnostic progress measure	Total variation between c_t and c_{t-1}.
u_t	Progress label for $B_{t-1} \to B_t$	$u_t \in \{0, 1\}$; 1 indicates recorded task progress.
$\mathcal{W}_t$	Window of recent validated transitions	$\mathcal{W}_t = \{t - K + 1, \ldots, t\}$.
$P_X,\ S_t,\ R_t$	Gap persistence, progress stagnation, and belief recurrence	Window-level signals used to estimate belief health.
W_τ^Δ	World state projected onto the active gap	Projects W_τ onto the current active gap Δ_t.
$d_J,\ d_W$	Jaccard distance and world state distance	d_W averages d_J over Entity, State, and Relation sets.
$\mathcal{H}_t$	Belief health score	$\mathcal{H}_t = 1 - \max_{X \in \{E,A\}}[P_X \max(S_t, R_t)]$.
$K,\ L$	Window size and maximum recurrence lag	Recurrence is evaluated over lags $1 \le \ell \le L < K$.
$\epsilon,\ \epsilon_W$	Diagnostic progress and state similarity thresholds	Used to identify confidence changes and recurring world states.
$\theta_H,\ \theta_R$	Health and cycle-detection thresholds	Trapping requires $\mathcal{H}_t \le \theta_H$; cycle diagnosis uses $R_t \ge \theta_R$.
Factorized diagnosis and recovery		
ρ_t	Diagnosed trapping dynamics	$\rho_t \in \{\text{Static}, \text{Cycle}, \text{Drift}\}$.
k_t^*	Dominant recurrence lag	The lag underlying R_t; cycle diagnosis additionally requires $k_t^* > 1$.
$X,\ X_t$	Gap type and its selection at step t	$X \in \{E, A\}$; X_t denotes the selected type(s) in Algorithm 1.
$\mathcal{C}_t$	Recovery constraints for action selection	Combines C_{pattern} and C_{gap}; empty during normal interaction.

B SUPPLEMENTARY METHOD DETAILS

B.1 END-TO-END WORKFLOW OF *PoS*

Algorithm 1 summarizes the end-to-end inference workflow of *PoS*. It follows the notation introduced in Section 3 and abstracts away implementation-specific details while preserving the main control flow and data dependencies of the framework. In particular, the Active Gap is maintained across consecutive steps until it is resolved, and trapping is assessed only after the current belief transition has been validated and incorporated into Belief Health. Any recovery constraint triggered by the resulting health state therefore guides action selection from the subsequent interaction step.

Algorithm 1 Overall Inference Workflow of *PoS*

Require: Task goal G, initial observation o_0, interaction budget T.
Require: Window size K, maximum lag $1 \leq L < K$, thresholds ϵ, ϵ_W, θ_H, and θ_R.
Ensure: Environment result or termination upon budget exhaustion.

$(B_0, \Delta_0) \leftarrow$ INITIALIZEBELIEF(G, o_0). ▷ $B_0 = (W_0, G, \Delta_0^E, \Delta_0^A)$
$\mathcal{C}_0 \leftarrow \emptyset$.
for $t = 0, 1, \ldots, T-1$ **do**
  $a_t \sim \pi(\cdot \mid B_t, \Delta_t, \mathcal{C}_t)$. ▷ $\mathcal{C}_t = \emptyset$ during normal interaction
  $o_{t+1} \leftarrow$ ENVIRONMENTSTEP(a_t).
  $\widetilde{B}_{t+1} \leftarrow \mathcal{U}(B_t, a_t, o_{t+1})$. ▷ Candidate belief update
  $B_{t+1} \leftarrow$ BELIEFSENTINEL$(B_t, \widetilde{B}_{t+1}, a_t, o_{t+1})$. ▷ Audit and revise before commitment
  if the task is diagnostic **then**
    $d_{t+1}^{\text{diag}} \leftarrow \frac{1}{2} \sum_x |c_{t+1}(x) - c_t(x)|$.
    $u_{t+1} \leftarrow \mathbb{I}\left[d_{t+1}^{\text{diag}} > \epsilon\right]$.
  else
    $u_{t+1} \leftarrow$ LABELEXECUTIONPROGRESS(B_t, B_{t+1}, Δ_t). ▷ Belief Sentinel judgment
  end if
  if ISTERMINAL(o_{t+1}) **then**
    return ENVIRONMENTRESULT.
  end if
  $\Delta_{t+1} \leftarrow$ ADVANCEACTIVEGAP(B_{t+1}, Δ_t). ▷ Keep Δ_t unless it is resolved
  $\mathcal{C}_{t+1} \leftarrow \emptyset$.
  if $t + 1 \geq K$ **then**
    $\mathcal{W}_{t+1} \leftarrow \{t - K + 2, \ldots, t + 1\}$.
    Compute P_E, P_A, S_{t+1}, and R_{t+1} over $\mathcal{W}_{t+1}$ using Eq. 5.
    Project each W_τ onto the current active gap Δ_{t+1}.
    $\mathcal{H}_{t+1} \leftarrow 1 - \max_{X \in \{E,A\}} [P_X \max(S_{t+1}, R_{t+1})]$.
    if $\mathcal{H}_{t+1} \leq \theta_H$ **then**
      $\rho_{t+1} \leftarrow$ CLASSIFYDYNAMICS$(\{B_\tau\}_{\tau \in \mathcal{W}_{t+1}}, \Delta_{t+1})$.
      Apply the STATIC, CYCLE, and DRIFT conditions in order.
      $X_{t+1} \leftarrow$ SELECTBLOCKEDGAPTYPE(P_E, P_A, B_{t+1}).
      Select the higher-persistence type among nonempty gap sets. On a tie, prioritize epistemic gaps for diagnostic tasks and achievement gaps for execution tasks.
      $\mathcal{C}_{t+1} \leftarrow \mathcal{C}_{\text{pattern}}(\rho_{t+1}, \mathcal{W}_{t+1}) \cup \mathcal{C}_{\text{gap}}(X_{t+1}, B_{t+1}, \Delta_{t+1})$.
    end if
  end if
end for
return BUDGETEXHAUSTED.

B.2 BELIEF TRAPPING DETECTION AND RECOVERY

Step-level progress assessment. For diagnostic tasks, $c_t(x) \in [0, 1]$ denotes the confidence assigned to a goal-relevant inferred State or Relation. These values are maintained independently without normalization across records. The progress measure in Eq. 4 aggregates absolute confidence changes between consecutive validated beliefs, so both increases and decreases contribute to the progress label when the aggregate change exceeds ϵ. For execution tasks, the Belief Sentinel evaluates progress relative to the active gap using the criteria in Section 3.3.

Window-based health estimation. Trapping detection begins once K validated transitions are available. At each step t, historical world states within $\mathcal{W}_t$ are projected onto the same current active gap Δ_t, selecting relevant records through their existing relations. The comparison underlying d_W includes confidence changes even when descriptive content remains unchanged. When the active gap changes, the existing window is retained and its historical world states are projected onto the new gap. The resulting persistence, stagnation, and recurrence signals are combined according to Eq. 5, with trapping detected when $\mathcal{H}_t \leq \theta_H$.

Factorized trapping diagnosis. Once trapping is detected, the STATIC, CYCLE, and DRIFT conditions are evaluated in order. If none is satisfied, *PoS* uses a generic recovery prompt rather than assigning a trapping pattern. The blocked gap type is selected by comparing persistence across nonempty gap sets. When persistence is equal, diagnostic tasks prioritize epistemic gaps, whereas execution tasks prioritize achievement gaps.

Recovery constraint composition. The diagnosed pattern and selected gap type are incorporated into a recovery prompt together with the relevant Entities and the active gap. The resulting constraints address both the unproductive behavior and the unresolved requirement, and guide subsequent action selection while the active gap remains unchanged. After each validated transition, *PoS* recomputes belief health and releases the constraints when $\mathcal{H}_t > \theta_H$. Otherwise, it updates the diagnosis and corresponding recovery constraints.

B.3 DESIGN RATIONALE FOR BELIEF TRAPPING

Choice of trapping signals. The three signals capture complementary aspects of unproductive interaction. Gap Persistence ties the assessment to requirements that remain unresolved, while Progress Stagnation measures the absence of recorded task progress. Belief Recurrence complements these signals by identifying repeated belief configurations that may occur despite positive step-level progress labels. Behavioral statistics such as action repetition or interaction count do not directly establish whether the task is advancing. Repeated tool calls may acquire relevant information, whereas different actions may leave the same requirement unresolved. Likewise, confidence levels alone do not indicate progress, as high confidence can persist during stalled investigation and low confidence can accompany useful evidence gathering. The selected signals instead connect unresolved requirements with changes and recurring patterns in the maintained belief.

Aggregation of trapping signals. We design the health score to capture stagnation or recurrence associated with persistent unresolved requirements, without treating gap persistence alone as sufficient evidence of trapping. Let $p_t = \max_{X \in \{E,A\}} P_X$ denote the largest gap persistence within the current window. A geometric mean can miss trapping because stagnation and recurrence need not occur together. Specifically, $\mathcal{H}_t^{\text{geo}} = 1 - (p_t S_t R_t)^{1/3}$ yields $\mathcal{H}_t^{\text{geo}} = 1$ whenever either S_t or R_t is zero, regardless of how high the other signals are. An agent may make no progress without revisiting earlier beliefs, or repeatedly return to earlier beliefs while confidence changes produce positive progress labels. In either case, the absence of one signal prevents the geometric mean from identifying trapping. By contrast, $\mathcal{H}_t^{\max} = 1 - \max(p_t, S_t, R_t)$ can incorrectly identify trapping solely because unresolved requirements persist. When $p_t = 1$, it yields $\mathcal{H}_t^{\max} = 0$ even if $S_t = R_t = 0$, meaning that every transition records progress and no belief recurrence is detected. A gap may remain unresolved throughout the window while the agent makes intermediate progress toward resolving it. The maximum formulation therefore confuses an unfinished requirement with stalled progress. *PoS* therefore uses

$$\mathcal{H}_t = 1 - p_t \max(S_t, R_t), \tag{8}$$

which weights the stronger stagnation or recurrence signal by the persistence of unresolved requirements. This preserves sensitivity to either form of unproductive belief evolution without treating gap persistence alone as sufficient evidence of trapping. Table 4 illustrates these distinctions through three boundary configurations.

Table 4: Health scores under three illustrative boundary configurations. Lower values indicate stronger trapping signals.

Boundary case	$(\boldsymbol{p_t}, \boldsymbol{S_t}, \boldsymbol{R_t})$	$\mathcal{H}_t^{\text{geo}}$	$\mathcal{H}_t^{\max}$	$\mathcal{H}_t$
Stagnation without recurrence	$(1, 1, 0)$	1	0	0
Recurrence despite positive progress labels	$(1, 0, 1)$	1	0	0
Progress without gap closure	$(1, 0, 0)$	1	0	1

C Benchmark Details and Evaluation Protocols

C.1 Task Taxonomy

For concise reporting, we organize each benchmark according to its native task structure or evaluation dimensions. ALFWorld and LOCA-Bench are grouped by task type, while RCA-100 and ClinDiag follow the diagnostic targets and dataset subsets defined by their respective benchmarks.

C.1.1 ALFWorld

- **Place**: Pick & Place; Pick Two & Place.
- **Transform**: Clean & Place; Heat & Place; Cool & Place.
- **Inspect**: Examine in Light.

C.1.2 LOCA-Bench

- **Cross-source Workflows**: Canvas Arrange Exam; Course Assistant; Apply PhD Email; Set Conference/CR Deadline; Payable Invoice Checker.
- **Structured Analytics**: A/B Testing; Academic Warning; Excel Market Research; Machine Operating; NHL B2B Analysis.
- **Environment Operations**: Canvas List Test; Filter Low-Selling Products; Update Material Inventory; WooCommerce New Welcome; WooCommerce Stock Alert.

C.1.3 RCA-100

- **Fault Triage (FT)**: Identify the type of failure responsible for the observed incident.
- **Root-Cause Localization (RCL)**: Locate the entity at which the root cause occurs.
- **Joint**: Require both the fault type and the root-cause entity to be correctly identified.

C.1.4 ClinDiag

- **Common**: Real-world clinical cases collected from published case reports across multiple medical specialties.
- **Rare**: Clinical cases covering diverse categories of rare diseases.

C.2 Dataset Statistics and Splits

Table 5 summarizes the evaluation splits and numbers of cases used in our experiments.

For ALFWorld, we use the standard `valid_unseen` split containing 134 tasks in previously unseen environments. The split contains 41 Place, 75 Transform, and 18 Inspect tasks, and the same cases are used for every method and backbone.

LOCA-Bench contains 15 seed tasks instantiated at seven environment-description lengths: 8K, 16K, 32K, 64K, 96K, 128K, and 256K tokens. For each task and target length, five random environment seeds generate distinct environment states while preserving the task definition. This produces 75 cases at each context length and 525 cases in total. Each of our three task categories contains five seed tasks and therefore 175 cases. The environment seeds control benchmark instantiation and should not be confused with repeated model runs.

Table 5: Dataset statistics and evaluation splits.

Benchmark	Evaluation Split	Task Category	# Cases
ALFWorld	`valid_unseen`	Place	41
		Transform	75
		Inspect	18
		Total	**134**
LOCA-Bench	Official benchmark	Cross-source Workflows	175
		Structured Analytics	175
		Environment Operations	175
		Total	**525**
RCA-100	Full benchmark	All diagnostic cases	103
ClinDiag	Sampled evaluation subset	Common	302
		Rare	302
		Total	**604**

For RCA-100, we evaluate all 103 cases in the released benchmark. Every case is scored on fault triage (FT), root-cause localization (RCL), and their joint correctness, so these targets share the same underlying set of diagnostic cases.

The complete ClinDiag benchmark additionally contains an access-controlled Emergency subset derived from MIMIC-IV-Ext, which was unavailable under our current data-access setting and is therefore excluded. Because the available Common pool is substantially larger than the Rare subset, we randomly sample 302 Common cases to match all 302 available Rare cases. This yields a balanced evaluation set of 604 cases, prevents the Common subset from dominating the aggregate result, and keeps the evaluation computationally tractable. The Common subset is sampled once before evaluation and is held fixed across all methods and backbones.

C.3 Interactive Evaluation Protocols

All methods interact with each benchmark through the same benchmark-specific interface. The context-management baselines operate within the shared ReAct harness, while LongHorizon-Harness retains its native Manage–Execute–Audit control structure.

- **ALFWorld.** The agent receives a natural-language goal and an initial textual observation. At each step, it executes one command from the environment's admissible action set and receives the resulting observation. A case succeeds when the environment confirms that the goal conditions have been satisfied; otherwise, it terminates when the action budget is exhausted.
- **LOCA-Bench.** The agent receives a task instruction and a case-specific set of MCP tools for inspecting and modifying the task environment. Each tool call constitutes one environment action, and its result becomes the next observation. The agent invokes the ``claim_done'' tool after completing the task, after which the official evaluator determines success; an uncompleted case terminates upon budget exhaustion.
- **RCA-100.** The agent initially receives the incident metadata, while metrics, logs, traces, events, alerts, and system topology must be acquired through diagnostic tools. Each tool call constitutes one environment action. The agent terminates the case by calling ``submit_root_cause'' with a fault type and a root-cause entity. We evaluate the two submitted fields separately and report joint accuracy when both are correct; cases without a valid submission before the action budget are treated as failures.
- **ClinDiag.** The agent initially observes only the patient's initial presentation. It acquires additional evidence by asking one specific question per action about medical history, physical examination, laboratory tests, radiographic tests, or other diagnostic procedures. The case terminates when the agent calls ``submit_diagnosis'', and the submitted diagnosis is evaluated against the reference diagnosis by the benchmark judge. Cases without a valid submission before the action budget are treated as failures.

Table 6: Evaluation metrics, interaction budgets, and termination conditions. All methods use the same budget within each benchmark.

Benchmark	Reported Metrics	Maximum Budget	Terminal Evaluation
ALFWorld	Category and Overall task success rates	50 text actions	The environment terminates the case when the goal is satisfied or another terminal state is reached. Success requires the environment's goal-completion signal.
LOCA-Bench	Category and Overall task success rates	100 MCP tool calls	Calling `claim_done` triggers the official evaluator. A terminal score of at least 1.0 is counted as success.
RCA-100	FT, RCL, and Joint accuracy	50 decision turns; at most 150 tool calls	A valid `submit_root_cause` call terminates the case. FT and RCL are scored separately, and Joint requires both to be correct.
ClinDiag	Diagnosis accuracy	30 tool calls	A valid `submit_diagnosis` call terminates the case, after which the submitted diagnosis is evaluated against the reference diagnosis.

C.4 Metrics, Budgets, and Termination

Let $y_i \in \{0, 1\}$ indicate whether case i is successfully completed. For a task category $\mathcal{C}$, we compute its success rate as

$$\mathrm{SR}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} y_i. \tag{9}$$

The Overall results for ALFWorld and LOCA-Bench are micro-averaged over all evaluated cases rather than averaged across category-level success rates. For LOCA-Bench, the main result pools cases across all environment lengths and random seeds, whereas the context-scaling analysis reports the same metric separately at each environment length.

For RCA-100, let y_i^{FT} and y_i^{RCL} indicate whether the submitted fault type and root-cause entity are correct, respectively. We report

$$\mathrm{FT} = \frac{1}{N} \sum_{i=1}^{N} y_i^{\mathrm{FT}}, \qquad \mathrm{RCL} = \frac{1}{N} \sum_{i=1}^{N} y_i^{\mathrm{RCL}}, \qquad \mathrm{Joint} = \frac{1}{N} \sum_{i=1}^{N} y_i^{\mathrm{FT}} y_i^{\mathrm{RCL}}. \tag{10}$$

Thus, Joint counts a case as correct only when both the fault type and the root-cause entity are correctly identified.

For ClinDiag, diagnosis accuracy is the proportion of cases whose submitted final diagnosis is judged clinically equivalent to the reference diagnosis. The judge accepts synonyms, standard abbreviations, and clinically equivalent formulations, while rejecting related but incorrect conditions or unsupported over-specific diagnoses. One valid judge verdict is used for each case; malformed judge outputs are retried up to two times.

A valid terminal submission ends the case regardless of whether its answer is correct. Malformed submissions are rejected with an error observation and may be corrected within the remaining budget. Other invalid actions or failed tool calls, including tool-level timeouts, likewise produce an error observation and consume the corresponding interaction opportunity without terminating the case. **A case that reaches its maximum budget without successful completion is counted as a failure and remains in the denominator.** Infrastructure failures outside the agent–environment interaction are retried and are not scored as task failures.

D Implementation and Reproducibility Details

D.1 Backbone and Agent Configuration

We evaluate all methods with three backbone models: `Qwen3.7-Plus`, `Kimi-K3`, and `GLM-5.3`. The model is accessed through the DashScope OpenAI-compatible API (`https://dashscope.aliyuncs.com/compatible-mode/v1`); no fixed API snapshot is available. For each backbone setting, all LLM-based components of *PoS* and baselines use the same backbone as the corresponding baseline agents. We do not use a stronger auxiliary model for belief construction, validation, or trapping-aware recovery. The Task Agent selects its action from the current task-conditioned belief state, which serves as its decision context.

The only model fixed independently of the evaluated backbone occurs in the medical diagnosis benchmark ClinDiag. Its provider and judge are treated as part of the benchmark environment rather than as components of the evaluated agent. To keep this environment identical across backbone settings, both are implemented with `Qwen3.7-plus` for all methods and backbones. All agent-side model calls use deterministic decoding with temperature 0. We set the maximum generation length to 512 tokens for ALFWorld, where each interaction primarily requires selecting an environment action, and to 4096 tokens for LOCA-Bench, RCA-100, and ClinDiag.

D.2 Baseline Selection

We select baselines to cover representative forms of decision context while remaining applicable under a common cross-domain evaluation setting. Raw Trajectory serves as the uncompressed reference and exposes the complete interaction history directly to the Agent. Among memory-based approaches, ACON (Kang et al., 2025), PACE (Wei et al., 2026), and HiAgent (Hu et al., 2025) represent complementary strategies based on history compression, adaptive context granularity, and hierarchical subgoal-oriented working memory, respectively. We further include LongHorizon-Harness (Ma et al., 2026) as a closely related state-oriented baseline because it explicitly maintains an audited task state across long-horizon execution. Together, these methods span raw history, compressed and selectively retained context, hierarchical memory, and explicit task-state maintenance, while remaining adaptable to the same four evaluation environments and backbone settings.

We do not attempt to exhaustively benchmark every method discussed in Section 2, as several approaches differ substantially in their training assumptions, control architecture, or task-specific state representation. For example, SUPO (Lu et al., 2026) jointly optimizes summarization and tool-use policies through reinforcement learning, while CaT (Liu et al., 2026) and AgentFold (Ye et al., 2026) rely on learned policies for actively manipulating the working context. Several belief-oriented methods are likewise specialized to particular reasoning structures or task families. In particular, Graph of States (Luo et al., 2026) is designed for abductive reasoning and couples belief representation with a causal graph and a state machine that explicitly constrain hypothesis evolution, making its representation closely tied to diagnosis-style tasks rather than a common decision context that can be applied unchanged across all four benchmarks. Similarly, Tru-POMDP (Tang et al., 2026) assumes structured hypotheses and belief-space planning for uncertain task planning. We therefore prioritize baselines that provide representative and practically comparable alternatives to *PoS* under a unified cross-domain evaluation protocol.

D.3 Baseline Implementation

Shared evaluation protocol. We evaluate all baselines through the same benchmark adapters, tool interfaces, task instructions, termination protocol, and environment-action budgets described in Appendix C. All LLM-based components follow the backbone configuration described in Appendix D.1. Because different policies naturally induce different interaction trajectories, the methods do not receive identical observations; instead, they share the same evidence-acquisition interface and differ in how the evidence produced by their own interactions is represented and exposed for subsequent decision making. Raw Trajectory, ACON, PACE, and HiAgent are integrated into a common ReAct harness (Yao et al., 2023). LongHorizon-Harness retains its native Manage–Execute–Audit control structure, while using the same benchmark adapters, tools, backbone, and environment-action budget.

Table 7: **Decision contexts and implementation provenance of the evaluated baselines.** All methods share the same benchmark interface and environment-action budget. The main differences lie in how interaction evidence is represented and exposed for subsequent decision making.

Method	Decision Context	Implementation Source	Material Adaptation
Raw Trajectory	The task specification, current observation, and complete chronological action–observation trajectory.	Shared ReAct implementation.	No compression, retrieval, or auxiliary context-management operation.
ACON	A cumulative compressed summary together with the three most recent interaction turns in their original form; before compression is triggered, the full trajectory is retained.	Official implementation (commit `d63f9ae18959`) (Kang et al., 2025).	Runtime compression is preserved, while failure-driven guideline optimization and compressor distillation are omitted.
PACE	A chronological sequence of historical chunks represented at full, detailed, brief, or placeholder granularity according to predicted relevance, with the two most recent chunks retained in full and selected chunks recoverable through glimpse.	Reimplemented from the published method and Algorithm 1 (Wei et al., 2026).	The published repository was inaccessible during implementation; `text-embedding-v4` replaces the original BGE-M3 encoder for relevance scoring.
HiAgent	Summaries of completed subgoals followed by the full action–observation trajectory of the active subgoal; original trajectories of completed subgoals remain retrievable when needed.	Official implementation (commit `cebdd8e4eace`) (Hu et al., 2025).	The hierarchical working-memory mechanism is ported from AgentBoard to the shared benchmark interface, with benchmark-level summary and folding parameters.
LongHorizon-Harness	An audited task state containing requirements, artifacts, and facts, together with a bounded subtask contract for the current execution round.	Official implementation (Ma et al., 2026).	The native Manage–Execute–Audit architecture is preserved, with its benchmark-facing tool interfaces adapted to our common evaluation setup.

Material implementation differences. The original baselines were developed for different environments and agent protocols, so their end-to-end runners cannot be directly applied to all four benchmarks. We preserve the defining context-management mechanism of each method while adapting its interaction layer to the common benchmark interfaces. All methods receive the same task-independent background information, tool documentation, and termination instructions. Method-specific prompts contain only the instructions required to construct or operate the corresponding context representation. When an original response format is incompatible with the benchmark tool interface, we provide format demonstrations containing only interface-level examples and no evaluated cases or target answers.

ACON. ACON learns compression guidelines from contrasting successful and failed trajectories and can further distill the optimized compressor (Kang et al., 2025). Our evaluation benchmarks do not provide dedicated training splits for learning such benchmark-specific compression policies, while optimizing them on evaluation cases would conflate inference-time context management with task-specific training. We therefore preserve ACON's runtime history-compression mechanism but use fixed benchmark-level preservation guidelines without failure-driven guideline optimization or compressor distillation. The guidelines are case-independent and specify only generic categories of information that should be retained. Compression thresholds and output budgets are fixed at the benchmark level and shared across all evaluated backbones.

PACE. PACE retains the interaction history in an external chunk store and exposes each chunk at an adaptive granularity determined by its predicted relevance to the next decision (Wei et al., 2026).

We preserve its four representation levels, relevance-based granularity selection, chronological ordering, and glimpse mechanism. Because the repository linked by the paper was inaccessible during our implementation, we reproduce the method from its published equations, default settings, and Algorithm 1. The original BGE-M3 encoder is replaced with `text-embedding-v4` for relevance scoring, while the evaluated backbone generates the textual detailed and brief representations.

HiAgent. HiAgent organizes working memory around Agent-generated subgoals (Hu et al., 2025). The trajectory associated with the active subgoal is retained in full, whereas completed subgoals are represented by summaries and their original trajectories remain available for retrieval. Since the official implementation is built on AgentBoard, we port the subgoal-based working-memory mechanism into the shared benchmark interface and convert its interaction protocol to the common structured action format. Summary-length and folding parameters are adapted at the benchmark level and then fixed across all cases and backbone models.

LongHorizon-Harness. LongHorizon-Harness differs from the other baselines in that it defines a complete Manage–Execute–Audit control architecture rather than only a context representation (Ma et al., 2026). At each round, the Manager constructs a bounded subtask from the audited task state, the Executor interacts with the environment to complete it, and the Auditor verifies the resulting state before verified requirements, artifacts, and facts are propagated to the next round. We base our implementation on the official LongHorizon-Harness codebase and preserve its native Manager–Executor–Auditor workflow and audited task-state representation. We only adapt the benchmark-facing tool interfaces and action formats so that the method can operate through the same evaluation interfaces as the other baselines. Any Auditor operation that queries the environment is counted against the same environment-action budget as an Executor action; purely internal Manager, Executor, or Auditor LLM calls are treated as computational overhead.

D.4 PoS Hyperparameters

Table 8 summarizes the hyperparameter configuration of *PoS*. Belief updates and local consistency audits are scheduled at every interaction step, while global audits are scheduled every 32 steps on LOCA-Bench and every 8 steps on the other benchmarks. The diagnostic progress threshold applies to RCA-100 and ClinDiag, whereas execution progress is assessed by the Belief Sentinel. Trapping detection begins once eight validated transitions are available, and recovery uses up to four recent transitions labeled as making no progress.

Table 8: Hyperparameter configuration of *PoS*. Intervals are measured in interaction steps.

Parameter	**Value**
Belief update interval	1
Local audit interval	1
Global audit interval (LOCA-Bench)	32
Global audit interval (other benchmarks)	8
Recent transitions supplied for local auditing	4
Detection window size K	8
Maximum recurrence lag L	4
Recurrence distance threshold ϵ_W	0.15
Diagnostic progress threshold ϵ	0.30
Health threshold θ_H	0.25
Cycle recurrence threshold θ_R	0.75
Maximum recent non-progress transitions for recovery	4

E Fine-Grained Main Results

Table 9 expands the aggregate results in Table 1 into task categories and diagnostic dimensions. This breakdown reveals which task requirements benefit from explicit belief maintenance, how validation and recovery contribute to different aspects of performance, and where challenges remain.

Table 9: Main results across four benchmarks (%). We report task success for ALFWorld and LOCA-Bench, failure triage (FT), root-cause localization (RCL), and joint accuracy for RCA-100, and diagnosis accuracy for ClinDiag. Env. Ops denotes Environment Operations.

Method	ALFWorld				LOCA-Bench				RCA-100			ClinDiag		
	Place	Transform	Inspect	Overall	Cross-source	Analytics	Env. Ops	Overall	FT	RCL	Joint	Common	Rare	Overall
Backbone: Qwen3.7-Plus														
Raw Trajectory	92.68	42.67	77.78	62.69	39.43	65.71	25.71	43.62	26.21	68.93	24.27	37.75	40.07	38.91
ACON	**97.56**	48.00	72.22	66.42	49.14	**75.43**	25.14	49.90	30.10	68.93	27.18	39.07	40.40	39.74
PACE	92.68	52.00	77.78	67.91	4.57	40.57	6.86	17.33	33.01	66.99	28.16	43.04	40.73	41.89
HiAgent	90.24	48.00	88.89	66.42	18.86	40.00	14.86	24.57	32.04	**76.70**	28.16	39.74	40.40	40.07
LongHorizon-Harness	95.12	56.00	88.89	72.39	53.71	72.57	31.43	52.57	31.07	75.73	26.21	41.72	39.40	40.56
PoS (OURS)	**97.56**	**81.33**	**100.00**	**88.81**	**58.86**	74.29	**36.00**	**56.38**	**39.81**	67.96	**38.83**	**48.01**	**42.05**	**45.03**
- Consistency Validation	92.68	65.33	66.67	73.88	42.29	69.14	22.29	44.57	33.01	66.99	31.07	47.35	41.72	44.54
- Trapping Diagnosis	85.37	70.67	55.56	73.13	48.00	71.43	25.14	48.19	35.92	63.11	33.98	45.03	39.74	42.38
Backbone: Kimi-K3														
Raw Trajectory	97.56	68.00	**100.00**	81.34	88.00	84.57	36.57	69.71	38.83	72.82	36.89	50.66	47.68	49.17
ACON	**100.00**	66.67	**100.00**	81.34	90.29	86.86	35.43	70.86	41.75	**79.61**	40.78	39.07	38.08	38.58
PACE	**100.00**	76.00	83.33	84.33	52.00	50.86	21.71	41.52	36.89	74.76	35.92	50.66	49.67	50.17
HiAgent	97.56	81.33	**100.00**	88.81	63.43	59.43	25.71	49.52	42.72	78.64	41.75	50.99	46.03	48.51
LongHorizon-Harness	**100.00**	86.67	94.44	91.79	84.57	80.00	38.29	67.62	43.69	76.70	38.83	51.32	**52.32**	51.82
PoS (OURS)	**100.00**	**89.33**	**100.00**	**94.03**	**92.57**	**89.14**	**41.14**	**74.29**	**51.46**	76.70	**51.46**	**57.28**	**52.32**	**54.80**
- Consistency Validation	97.56	88.00	94.44	91.79	84.57	81.71	36.00	67.43	48.54	75.73	48.54	56.95	51.99	54.47
- Trapping Diagnosis	97.56	82.67	94.44	88.81	88.00	85.14	35.43	69.52	46.60	72.82	46.60	52.98	48.68	50.83
Backbone: GLM-5.3														
Raw Trajectory	97.56	89.33	**100.00**	93.28	80.00	76.00	32.57	62.86	34.95	71.84	32.04	50.00	43.71	46.85
ACON	97.56	92.00	**100.00**	94.78	85.14	81.14	31.43	65.90	33.98	77.67	33.98	49.01	43.38	46.19
PACE	97.56	92.00	**100.00**	94.78	48.00	47.43	19.43	38.29	39.81	**78.64**	37.86	47.68	39.07	43.38
HiAgent	**100.00**	90.67	**100.00**	94.78	59.43	54.86	23.43	45.90	43.69	74.76	40.78	49.34	40.40	44.87
LongHorizon-Harness	97.56	93.33	**100.00**	95.52	81.71	78.29	33.14	64.38	38.83	73.79	31.07	48.68	42.38	45.53
PoS (OURS)	**100.00**	**94.67**	**100.00**	**97.01**	**88.00**	**85.71**	**38.86**	**70.86**	**50.49**	75.73	**49.51**	**54.97**	**49.34**	**52.15**
- Consistency Validation	97.56	93.33	94.44	94.78	79.43	75.43	33.14	62.67	49.51	74.76	47.57	53.97	49.01	51.49
- Trapping Diagnosis	97.56	89.33	94.44	92.54	85.14	79.43	32.57	65.71	45.63	71.84	42.72	50.99	46.03	48.51

(1) Explicit belief particularly benefits state tracking and diagnostic disambiguation. A key benefit of explicit belief is that observations from different steps jointly inform an updated understanding of the current world. By maintaining this understanding together with unresolved gaps, *PoS* connects what the agent has already established to what its next action must accomplish. This is particularly useful when successful execution depends on preserving state changes across actions, or when diagnosis requires combining several signals to distinguish competing hypotheses. The concentration of ALFWorld gains in `Transform` supports this interpretation, with `Qwen3.7-Plus` success increasing from 56.00% to 81.33%, while `Place` matches the strongest baseline at 97.56%. Our RCA-100 trace analysis further shows how belief integrates "request patterns" and "resource-utilization" signals around an identified entity, allowing the agent to rule out failure types that conflict with the accumulated observations. Most failure-type corrections occur in cases where Raw Trajectory already identifies the correct entity but misclassifies its failure, including 14 cases with `Kimi-K3` and 16 with `GLM-5.3`. Together with the trace analysis, this highlights how explicit belief consolidates failure characteristics around an entity, while the Active Gap directs further investigation toward evidence that distinguishes competing hypotheses.

(2) Validation and recovery support different aspects of diagnostic performance. On RCA-100, removing Consistency Validation reduces RCL by only 0.97 points under each backbone, but lowers FT by 6.80 points with Qwen3.7-Plus. This contrast highlights the importance of validating inferred fault states even when the agent can still locate the incident. Removing Trapping Diagnosis and its associated recovery has a larger effect on localization, reducing RCL by 3.88–4.85 points. The distinction also appears in ClinDiag's `Rare` subset, where removing validation lowers accuracy by 0.33 points under each backbone, whereas removing recovery causes losses of 2.31–3.64 points. These results show that the two mechanisms address different needs within diagnosis. Validation checks whether belief updates remain consistent and supported, while recovery redirects investigation when the current course of interaction no longer resolves diagnostic uncertainty.

(3) Better decision context does not replace execution skills or domain knowledge. Explicit belief construction and maintenance improve how task information is represented and updated, but successful completion also depends on the model's ability to act on that information. Although *PoS* achieves the highest success on LOCA-Bench's `Env.Ops` category under every backbone, performance remains between 36.00% and 41.14%. These tasks also demand reliable tool use and action sequencing, as identifying the required state change is different from executing the operations that produce it. Our ClinDiag trace analysis reveals a related limitation, with many incorrect diagnoses arising after the agent receives correct information but lacks the medical knowledge needed to interpret it. Even an accurate representation of the current World State is insufficient when the agent lacks the execution skills or domain knowledge required to complete the task. Addressing these limitations requires improvements in tool-use competence and domain-specific reasoning alongside belief maintenance, rather than relying on context management alone.

F Case Study

We use RCA-100 case `t024` with `Kimi-K3` to illustrate *PoS*'s trapping diagnosis and recovery. The agent observes abnormal inventory-service behavior but remains unable to distinguish database waiting from JVM-side processing despite continued investigation. *PoS* identifies "Static Stagnation" with a persistent epistemic gap and combines constraints against ineffective repetition with a requirement to seek evidence that distinguishes these explanations. The agent subsequently obtains CPU observations and integrates them with the earlier GC evidence, revising its belief and ultimately submitting the correct diagnosis. The case below traces this process from the unresolved diagnostic question to targeted recovery and belief revision.

RCA-100: recovering from stalled diagnosis **Case `t024` — Kimi-K3**

The agent must identify the fault entity and failure type. This excerpt covers one of three recovery episodes in a 28-action run. Step indices are zero-based.

1. Investigation continues, but the diagnostic question remains unresolved

Current belief. Slow inventory requests coincide with elevated garbage-collection (GC) activity.

Active epistemic gap. Are the delays caused by database waiting or processing inside the inventory JVM? Further log searches do not distinguish these explanations.

Trapping diagnosis. At step 22, the same gap persists with little change in the belief. PoS identifies *Belief Trapping* pattern as **Static Stagnation + Epistemic Gap**.

2. Factorized recovery redirects investigation toward the unresolved gap

Pattern constraint. Avoid repeating the ineffective action under the unchanged belief.

Gap constraint. Seek evidence that distinguishes database waiting from processing inside the JVM.

Agent-selected investigation. The agent chooses to inspect CPU activity. The recovery constraints specify what the investigation must resolve, rather than prescribe a tool call or provide the diagnosis.

3. New evidence changes the belief and advances diagnosis

New evidence. Recovery initially produces three further low-progress actions. Subsequent CPU queries reveal increased activity during the incident.

Belief revision. The agent combines the CPU evidence with the earlier GC observations, increases its support for JVM-side processing, and reduces its support for database waiting. The targeted epistemic gap is recorded as resolved at step 26.

Outcome. At step 27, the agent submits **`inventory / memoryPressure`**, matching both the entity and failure type in the benchmark label.

Scope. The case illustrates how targeted recovery restores diagnostic progress. CPU evidence supports the recorded belief revision but does not uniquely determine the final failure type.

G LIMITATIONS AND FUTURE DIRECTIONS

Two limitations motivate future work on the efficiency of belief maintenance and the reliability of knowledge-dependent reasoning in PoS.

(1) Cost of explicit belief maintenance. The performance gains of *PoS* come with additional token and computational costs. Its natural language form structured Belief States require repeated text processing and generation during maintenance and subsequent reasoning. Repeatedly expressing and revising beliefs in language introduces redundancy and may cause information omission or semantic distortion. Future work could explore task-conditioned **latent world-state representations** that encode decision-relevant information more compactly and support incremental updates without repeatedly expanding the state into text. The key is to encode information about the current world state in a latent representation with high information density that can be efficiently propagated and updated within the model. We aim to preserve an accurate understanding of the current world state while substantially reducing the token and computational costs of belief maintenance.

(2) Dependence on domain and task knowledge. Although *PoS* receives external observations through tools, it does not systematically integrate external domain-knowledge retrieval into belief construction and reasoning. Its ability to identify relevant evidence and interpret observations therefore depends largely on the backbone model's existing knowledge. Insufficient knowledge may lead to omitted facts or incorrect causal and diagnostic judgments, even when the resulting Belief State is internally consistent. Future work could incorporate **domain-knowledge retrieval** into belief updates and evidence interpretation, helping the agent determine what information to retain and how it supports or contradicts competing explanations. By incorporating relevant domain knowledge, we aim to improve the accuracy of Belief States and the reliability of subsequent reasoning in specialized tasks.