# COMPASS: Enhancing Agent Long-Horizon Reasoning with Evolving Context

Guangya Wan[*1,2], Mingyang Ling[1], Xiaoqi Ren[1], Rujun Han[1], Sheng Li[2], and Zizhao Zhang[1]

[1]Google Cloud AI
[2]University of Virginia

**Abstract**

Long-horizon tasks that require sustained reasoning and multiple tool interactions remain challenging for LLM agents: small errors compound across steps, and even state-of-the-art models often hallucinate or lose coherence. We identify *context management* as the central bottleneck—extended histories cause agents to overlook critical evidence or become distracted by irrelevant information, thus failing to replan or reflect from previous mistakes. To address this, we propose **COMPASS** (*Context-Organized Multi-Agent Planning and Strategy System*), a lightweight hierarchical framework that separates tactical execution, strategic oversight, and context organization into three specialized components: (1) a *Main Agent* that performs reasoning and tool use, (2) a *Meta-Thinker* that monitors progress and issues strategic interventions, and (3) a *Context Manager* that maintains concise, relevant progress briefs for different reasoning stages. Across three challenging benchmarks—GAIA, BrowseComp, and Humanity's Last Exam—COMPASS improves accuracy by up to 20% relative to both single- and multi-agent baselines. We further introduce a test-time scaling extension that elevates performance to match established DeepResearch agents, and a post-training pipeline that delegates context management to smaller models for enhanced efficiency.

## 1 Introduction

Large Language Model (LLM) agents have achieved impressive performance on tasks with well-defined reasoning paths and objectives (Comanici, 2025; Minaee et al., 2025). The emerging challenge for autonomous systems is mastering *long-horizon tasks* (LHT)—problems that demand sustained reasoning across multiple tool interactions while maintaining strategic coherence and adapting to unexpected outcomes (Sun et al., 2023; Xi et al., 2025). For example, a query from the BrowseComp dataset (Wei et al., 2025) may ask to identify a soccer player with specific yellow-card patterns across halves in a given year, requiring database lookups, referee validation, and integration of timing and match data.

Such tasks are challenging because agents must maintain a high success rate at **each step** of tool use. Minor errors—such as ambiguous search results or faulty API calls—can cascade, turning recoverable mistakes into systematic failures (Zhang et al., 2023). Hallucinations in LLMs are hard to avoid given current architectures (Xu et al., 2025), and even the most capable closed-source models struggle to sustain coherent plans over extended horizons (Gonzalez-Pumariega et al., 2025). Thus, effective long-horizon reasoning demands not only accurate tool use and instruction following capabilities, but also a proactive, strategic mindset to adaptively reflect and replan for more reliable outputs (Erdogan et al., 2025).

To address the challenges, **Single-agent systems (SAS)**, pioneered by the ReAct paradigm (Yao et al., 2023), operate in a think–act–observe loop until a final answer is produced. Their strength lies in fluid, end-to-end control, where a single model manages reasoning, planning, and tool use, often augmented by post-trained, tool-integrated reasoning models (Comanici, 2025; Li et al., 2025b) or inference-time "thinking" modules (Anthropic, 2025). However, SAS are limited by a unified context window—as trajectories lengthen, evidence may be truncated or misweighted, leading to premature conclusions and superficial self-reflection (Ding et al., 2024; Chakraborty et al., 2025; Hong et al., 2025; Liu, 2025).

In contrast, **multi-agent systems (MAS)** distribute tasks among specialized agents through decentralized handoffs or hierarchical coordination (Liang et al., 2024; Wu et al., 2024; Tran et al., 2025). These systems achieve state-of-the-art results on challenging agentic benchmarks (Huang et al., 2025; Wei et al., 2025), benefiting from explicit separation of context and roles—dedicated agents for planning, reflection, and execution that collectively enhance strategic reasoning (Tran et al., 2025). In practice, **human-in-the-loop (HITL)**

[*]Work done during internship at Google.



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3360–3380

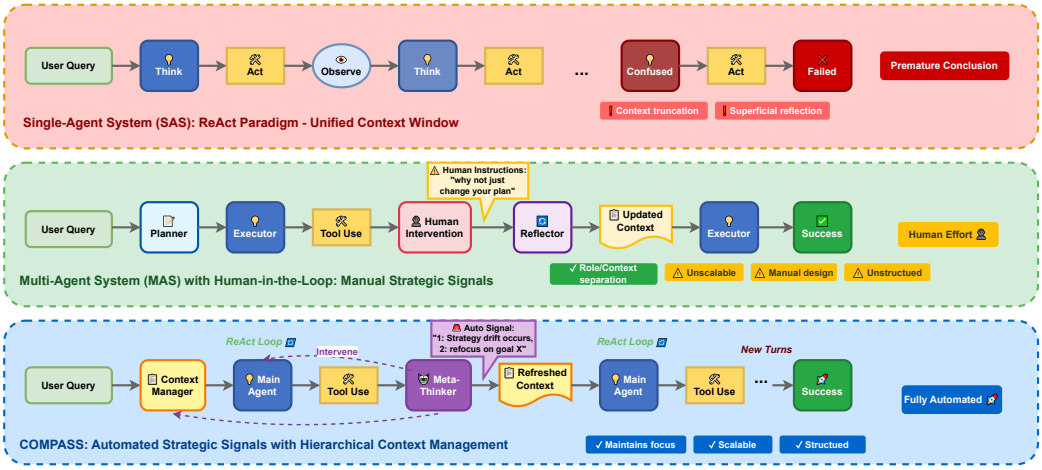


Figure 1: **Motivation for COMPASS.** ReAct-style SAS accumulate full dialogue histories, leading to context exhaustion and performance degradation. MAS with Human-in-the-loop improves performance through human's feedback but require human effort and lack scalability. COMPASS introduces automated monitor and context management agents that track reasoning, organize structured context, and maintain reliability with full automation.

interventions are often integrated into MAS (Takerngsaksiri et al., 2025), where specific workflows are designed to allow human operators to pause execution, alter context, or inject strategic signals (Figure 1) (Li et al., 2025a). While effective, these interventions rely on direct human feedback and manual design of when and how to inject signals, which are inherently unscalable and unreliable to align with the agent's evolving reasoning context.

While SAS are generally preferred for their fully autonomous nature—enabling greater extensibility and generalizability—existing context management techniques within SAS (Ding et al., 2024; Hong et al., 2025) still lack the explicit role and context separation that MAS architectures provide. To preserve the architectural advantages of MAS—dedicated strategic reasoning agents with isolated context—while maintaining the simplicity and autonomy of SAS, we introduce **COMPASS** (*Context-Organized Multi-Agent Planning and Strategy System*), a hierarchical framework that decouples tactical execution from strategic oversight through three collaborative components: a *Main Agent* that performs ReAct-style reasoning under dynamically refreshed context, a *Meta-Thinker* that asynchronously monitors the main agent's progress and issues strategic interventions, and a *Context Manager* that compresses full histories among turns into concise, structured briefs to maintain organized context flow across reasoning stages for stable long-horizon performance.

We evaluate COMPASS on challenging academic and deep-research benchmarks, introduce a test-time scaling extension (COMPASS-TTS), and present the CONTEXT-12B model for efficient context management. In sum, our contributions are: (1) **Formalizing strategic reasoning and context for LHT.** We formalize long-horizon strategic reasoning and highlight its dependence on explicit context management. (2) **The COMPASS framework.** We propose a hierarchical architecture that separates tactical execution from strategic oversight through explicit role and context separation, enabling reliable operation in error-prone long-horizon settings. (3) **Comprehensive evaluation and extensions.** We demonstrate COMPASS's effectiveness across benchmarks, with analyses and practical guidelines for scalable LLM agents.

## 2 Agents in Long-Horizon Tasks

**LLM agents.** We define an **LLM agent** as an autonomous system that leverages a large language model to iteratively reason, act, and observe in service of a goal. At each step $t$, the agent's behavior is conditioned on a context $C_t$, which comprises two components: a **static context** $C^{\text{static}}$, containing fixed information such as the initial query and tool specifications, and a **dynamic context** $C_t^{\text{dyn}}$, which accumulates execution traces including thoughts, tool calls, and observations. A single-agent system evolves this context through a thought–action–observation loop:

$$\text{SINGLEAGENT} : C_0 \xrightarrow{(r_0,a_0,o_0)} C_1 \xrightarrow{\cdots} \text{Answer},$$

while a *multi-agent system (MAS)* distributes the task across several coordinated agents, each with

Table 1: **Illustrative outcomes of meta-thinking decisions.** Each block shows correct and incorrect choices under two common scenarios of long horizon reasoning. Specific case studies are presented in Appendix §E.

| **Scenario 1: Handling Execution Failures** | | |
|---|---|---|
| **Ground Truth \ Decision** | **Continue** | **Revise** |
| **Local Error** | Correct (Persist) | Incorrect (Unnecessary Revision) |
| *Example: Search query too narrow* | *Refine terms* | *Abandon entire approach* |
| **Global Dead-End** | Incorrect (Persistence) | Correct (Revision) |
| *Example: API permanently broken* | *Retry broken API* | *Switch to new source* |

| **Scenario 2: Deciding Completion** | | |
|---|---|---|
| **Ground Truth \ Decision** | **Conclude** | **Continue** |
| **Correct Solution** | Correct (Conclude) | Incorrect (Overthinking) |
| *Example: Verified optimal solution* | *Return result* | *Keep exploring* |
| **Incorrect Solution** | Incorrect (Premature Stop) | Correct (Recovery) |
| *Example: Sign error in calculation* | *Submit wrong answer* | *Re-check computation* |

its own context $C_t^{(i)}$, orchestrated through mechanisms such as hierarchical control or peer-to-peer handoffs:

$$\textsc{MultiAgent} : \{C_t^{(1)}, \ldots, C_t^{(n)}\} \xrightarrow{\mathcal{F}} \text{Answer}.$$

**Long-horizon tasks.** We define a task as **long-horizon** if its successful completion requires a substantial sequence of interdependent reasoning and action steps (e.g., $> 10$), often involving iterative tool use, synthesis of intermediate results, and dynamic revision of plans. The principal challenge lies in managing the dynamic context: as the execution trace grows—often linearly with time, $|C_t^{\text{dyn}}| \propto O(t)$—it can exceed the model's finite context window, obscuring earlier but potentially essential information.

**Plans.** Long-horizon tasks typically begin with an initial *plan*, either implicitly generated by the agent or explicitly provided in the context (Huang et al., 2024b). Formally, a plan is a sequence of steps $(s_1, s_2, \ldots, s_T)$, where each $s_i$ specifies a concrete action or guideline. Such plans provide a useful backbone but are rarely sufficient on their own (Sun et al., 2023): unexpected tool responses or initial oversights often require adaptation.

**Tactical reasoning.** Given the current step $s_i$ from planning and the dynamic context $C_t$, tactical reasoning determines how to execute $s_i$ to produce useful outputs such as reasoning traces $r_t$ or tool responses $o_t$:

$$(r_t, o_t) = f_{\text{tac}}(s_i, C_t).$$

This process assumes the current plan remains valid, and focuses on accurate instruction following and step-level tool execution to achieve the tasks.

**Strategic reasoning.** Conditioned on the evolving context $C_t$, strategic reasoning monitors past reasoning for anomalies or inconsistencies and determines whether adjustments are required for subsequent planning:

$$(s_1, \ldots, s_T)' = f_{\text{strat}}(s_{1:i}, C_t).$$

If no anomaly is detected, execution continues as planned; if issues are identified, the next trajectory is revised to incorporate corrections or new information, such as an additional verification or backtracking to a earlier stage. If the reasoning already supports a sufficient solution, the process terminates with a final `<answer>`. Therefore, Long-horizon performance depends on the interplay between *tactical precision* at each step and *strategic oversight* across steps. Tactical reasoning ensures faithful local execution, while strategic reasoning governs when to correct, adapt, or conclude. Table 1 illustrates some confusion cases, and full examples are covered in in Appendix §E.

## 3 Methods

### 3.1 Architectural Overview

At the core of COMPASS are three specialized agents with clearly separated responsibilities:

**Main Agent** serves as the primary executor for *tactical reasoning* with a ReAct-style workflow. At step $t$, it alternates between generating an intermediate thought and producing a tool command appended to the running trace and iteratively continues until the desired answer is obtained. Viewed in isolation, this resembles the single-agent system defined in Section 2. Within COMPASS, however, the Main Agent is supplied with a *renewed context* to follow from the Context Manager whenever a strategic intervention occurs and can be *interrupted* at any moment by the meta-thinking agent.

**Meta-Thinker** *reasons over previous reasoning*, monitoring the reasoning trajectory together with

**Algorithm 1** COMPASS: Dual-Loop with Meta Oversight and Context Management.

**Require:** Query $q$, tools $\mathcal{O}$, agents $\mathcal{A}^{\text{main}}, \mathcal{A}^{\text{meta}}, \mathcal{A}^{\text{ctx}}$, max iterations $T_{\max}, I_{\max}$
**Ensure:** Solution $y^*$ for query $q$

$n_0 \leftarrow \mathcal{A}^{\text{meta}}.\text{Initialize}(q)$ ▷ Initial knowledge and planning as *notes*
**for** $t = 0, 1, \ldots, T_{\max} - 1$ **do** ▷ Outer loop of strategic reasoning and context update
  $x_t \leftarrow \mathcal{A}^{\text{ctx}}.\text{SynthesizeContext}(n_t, q)$
  $\mathcal{T}_t \leftarrow \text{ExecuteTurn}(\mathcal{A}^{\text{main}}, \mathcal{A}^{\text{meta}}, x_t, I_{\max})$ ▷ ReAct loop with monitoring; see Alg. 2
  $\textit{decision} \leftarrow \mathcal{A}^{\text{meta}}.\text{GetDecision}(\mathcal{T}_t, x_t)$ ▷ Strategic decision from meta-thinking
  **if** $\textit{decision} = \text{STOP}$ **or** $t = T_{\max} - 1$ **then**
    **return** $\text{ExtractAnswer}(\mathcal{T}_t, n_t)$
  **end if**
  $n_{t+1} \leftarrow \text{UpdateNotes}(n_t, \mathcal{T}_t, \textit{decision})$
**end for**

**ExtractAnswer**: re-summarizes the final trace $\mathcal{T}_t$ with context from notes $n_t$ to produce the `<answer>` (see App.§ B.2).
**UpdateNotes**: appends the current context in a structured form to the rolling set of prior briefs $\{x_0, \ldots, x_{t-1}\}$ (see App.§ B.2).

its dedicated context. It remains silent until anomalies are detected—such as looping behavior, tool misuse, or signs of task completion. Once triggered, it issues a high-level strategic signal to prompt reflection, termination, or verification, which is then passed to the Context Manager. Running asynchronously ensures that these interventions do not block the Main Agent's reasoning, thereby preserving execution fluidity.Because it operates on a single turn rather than full traces, the Meta-Thinker is designed to be lightweight, capable of catching up with the Main Agent's operation with low latency, especially when combined with prompt-caching techniques (Gim et al., 2024). This design also enables a significantly higher context caching rate compared to single-agent baselines, reducing effective API costs despite the multi-agent overhead (see Appendix D).

**Context Manager** serves as the system's adaptive context controller, responsible for determining what information should enter the active context for Main and Meta-thinking agent at each iteration. It *synthesizes a new, task-specific context* by selectively drawing from three sources: (i) the persistent *structured notes* accumulated across turns, (ii) the current reasoning trajectory from Main Agent, and (iii) the Meta-Thinker's strategic signal. Through this process, the Context Manager provides the Main Agent with a concise and relevant context that preserves continuity while avoiding redundancy or distraction. After each iteration, the resulting refreshed context is also appended to the evolving note store for future retrieval, ensuring long-horizon coherence with minimal memory overhead.

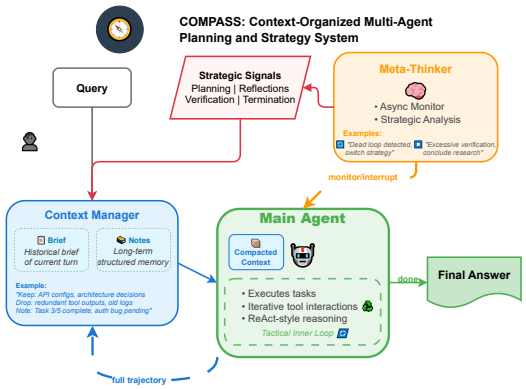


Figure 2: **The COMPASS dual-loop framework.** The *Main Agent* performs tool interactions by following the instructions from continually refreshed context; the *Meta-Thinker* asynchronously monitors trajectories and triggers strategic decisions, and the *Context Manager* compresses full histories (from structured notes) into concise, contextual aware briefs back to Main Agent.

### 3.2 Trajectory Lifecycle

The COMPASS framework (Algorithm 1) begins when the user issues a query $q$, which initializes the working state $h_0 = (q)$ via the meta-thinker. Before execution, the Meta-Thinker performs an *initial planning step*, outlining a coarse action sequence or information sources to explore. This plan defines the first context $x_0$, serving as a reference for subsequent meta-decisions. Execution then proceeds through iterative outer loops of reasoning and oversight, each consisting of two coupled stages:
**(1) Reasoning, Action, Monitoring, and Reflection.** During each iteration $t$, the Main Agent performs *tactical reasoning and action* using the current context $x_t$, producing intermediate thoughts and tool invocations that expand the trajectory $\mathcal{T}_t$.

Concurrently, the Meta-Thinker engages in *strategic monitoring*, asynchronously inspecting $\mathcal{T}_t$ for anomalies (e.g., loops, tool misuse, reasoning drift) or completion signals. Upon detection, it issues a high-level strategic signal (e.g. replan or verify) that guides the next context synthesis.

**(2) Context Refresh and Note Integration.** After a meta interruption, the Context Manager synthesizes a refreshed context $x_{t+1} = \text{SynthesizeContext}(q, \mathcal{T}_t, n_t, \textit{decision}_t)$ by integrating verified evidence, relevant notes (raw query $q$ and initial plan are included if no dynamic context yet available), and the Meta-Thinker's signal $\textit{decision}_t$. It then appends extracted segments—key observations of what has worked, what has failed, and remaining uncertainties—to the note store (App.§ B.2), updating $n_{t+1} = n_t \cup \text{ExtractSections}(x_t)$ for future iterations.

If $\textit{decision}_t = \text{STOP}$, the Context Manager invokes the Answer Synthesizer (App.§ B.2) to generate the final output; otherwise, $(x_{t+1}, n_{t+1})$ are passed forward to initialize the next iteration. This loop continues until convergence or the maximum round $T_{\max}$ is reached, ensuring that long-horizon reasoning proceeds with bounded context size and persistent continuity across turns.

Table 2: Pass@1(%) are reported. Datasets:**BC**=*BrowseComp*, **GAIA**, **HLE**. Tools: Search, Coding, Terminal, Browsing, Thinking (for thinking over previous reasoning), Context (for keeping the most relevant context).

| Method | Tools | BC | GAIA | HLE |
|---|---|---|---|---|
| **Gemini 2.5 Pro** | | | | |
| SINGLE-AGENT | | | | |
| Search and Browse Only | | 16.8 | 58.6 | 14.8 |
| + Meta-Thinking | | 26.4 | 61.6 | 20.6 |
| + Context | | 29.8 | 63.1 | 28.4 |
| MULTI-AGENT | | | | |
| Agent-as-a-Tool | | 31.8 | 65.5 | 28.6 |
| Decentralized Handoffs | | 28.1 | 64.8 | 28.3 |
| Iterative Refinement | | 30.5 | 65.3 | 27.9 |
| **COMPASS** | | **35.4** | **67.8** | **31.7** |
| **Gemini 2.5 Flash** | | | | |
| SINGLE-AGENT | | | | |
| Search and Browse Only | | 12.1 | 53.5 | 12.2 |
| + Meta-Thinking | | 19.8 | 56.3 | 18.7 |
| + Context | | 22.6 | 58.6 | 22.9 |
| MULTI-AGENT | | | | |
| Agent-as-a-Tool | | 23.4 | 58.9 | 22.1 |
| Decentralized Handoffs | | 21.9 | 58.1 | 23.8 |
| Iterative Refinement | | 22.8 | 58.7 | 23.3 |
| **COMPASS** | | **26.1** | **60.2** | **24.6** |
| ***Established Research Agents*** | | | | |
| DeepResearch (o3) | | **51.1** | 67.4 | 26.6 |
| DeepSeek V3.1 Terminus Agent | | 38.5 | 63.1 | 21.7 |
| Test Time Diffusion (Gemini 2.5 Pro) | | — | 69.1 | 33.9 |
| **COMPASS-TTS** (Gemini 2.5 Pro) | | 43.7 | **72.1** | **35.2** |

## 4 Experimental Results

**Benchmarks and Baselines.** Our evaluation for COMPASS focuses on **DeepResearch-style long-horizon benchmarks**, typically demanding 20+ reasoning–action steps: (i) **GAIA** (Mialon et al., 2024), including all Level 1–3 non-image tasks; (ii) **BrowseComp** (Wei et al., 2025), with 1,266 web navigation tasks requiring verification of entangled facts; and (iii) **Humans' Last Exam (HLE)** (Phan, 2025), yielding 2,158 questions across mathematics, humanities, and natural sciences after excluding image-based items. We compare COMPASS against two baseline groups: fundamental paradigms including single-agent systems (Search/Browse tools, +thinking Tool, +Context management tool), multi-agent systems (Manager hierarchical delegation and Decentralized Handoffs coordination with same set of agent as a tool), and Iterative Refinement workflows (Wang et al., 2023a)); Established research agents (OpenAI's DeepResearch, DeepSeek's Agent V3.1, Google's TestTime Diffusion) are included to compare with the Test-time scaling option (See Section 5.2). All experiments use **Gemini 2.5 Pro/Flash** and as backbone reasoning models and native Google serach/browsing and code execution tool is used.

**Evaluation Metrics.** Our primary metric is **Pass@1 accuracy**. We also assess strategic reasoning through four trajectory-level metrics, motivated by the failure mode in Table 1: **Persist Appropriateness Rate (PAR)** measures whether agents appropriately continue with valid plans, **Pivot Recognition (PVR)** captures whether agents pivot when current approaches fail, **Conclude Accuracy (CA)** indicates whether agents correctly recognize when to halt with a solution, and **Error-Recovery Continuation (ERC)** reflects whether agents continue searching after incorrect answers. These metrics, evaluated via LLM-as-a-Judge with structured outputs (App.§ B), expose precision-recall trade-offs in long-horizon problem solving—high PAR without PVR indicates blind adherence to failing plans, while high CA without ERC leads to premature termination—revealing more fine-grained details on how agents navigate critical decision points.

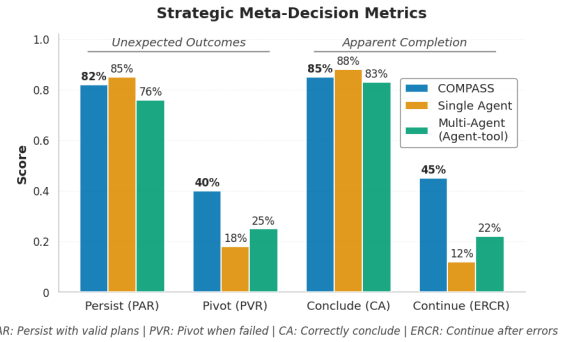


Figure 3: **Strategic meta-decision metrics across agent variants.** Bars report scores for four metrics to measure streatgic reasoning: Persist (PAR) and Pivot (PVR), and Conclude (CA) and Continue (ERCR); see Table 1 for examples and formal definitions.

### 4.1 Main Results.

Our main results (Table 2) reveal a clear path to enhancing agent capabilities. We first establish the importance of **structured reasoning and context**, as augmenting a single agent with meta-thinking and context management capabilities consistently improves its performance. Building on this, our **COMPASS** architecture further amplifies these gains by externalizing these roles into **dedicated agents for explicit monitoring, thinking, and context curation**—a method that shown more effectiveness than taking these functions as tools in single agent. Finally, these architectural benefits deliver **consistent performance** improvements across different models and benchmarks, with gains most pronounced on BrowseComp, where tasks require sustained multi-source navigation in long horizon, achieving results that are comparable to well-established deep-research agents with our framework when scaled with sampling.

**Strategic Reasoning and Case Analysis.** Figure 3 reveals how architectural separation improves strategic behavior. Single-agent baselines exhibit high PAR but low PVR—*blind persistence* where constraints become buried under accumulated outputs. Multi-agent baselines improve pivoting but sacrifice persistence, suggesting coordination alone is insufficient without context curation.

COMPASS balances these through complementary mechanisms: the Meta-Thinker detects anomalies (looping, tool misuse) before error cascade, while the Context Manager maintains concise briefs that preserve constraints. **Case studies (Appendix §E) demonstrate this prevents two failure modes common in Re-Act style agentic system—context overload causing premature conclusions (§E.4) and contextual pollution causing repeated errors (§E.1)—while enabling appropriate persistence during refinement (§E.1) and strategic pivoting at dead ends (§E.2).**

**Ablation Studies.** Table 3 systematically ablates Meta-Thinking and Context Management components on BrowseComp, revealing distinct failure modes and scaling behaviors. Removing the Meta-Thinker entirely collapses adaptive capability, causing *blind persistence* with high task completion but near-zero strategic metrics. Scaling up the Meta-Thinker shows systematic improvements in performance and strategic reasoning while maintaining token efficiency, indicating that *oversight quality* matters more than raw capacity. Context Manager ablations reveal different trade-offs: removing it causes token bloat from repeatedly revisiting failed attempts, while certain configurations like Gemini 2.5 Flash exhibit a failure mode of *excessive plan revision*, triggering strategic interventions that extend execution while improving adaptability. This highlights that context curation requires balancing compression with strategic signaling rather than pure summarization. Neither component alone achieves the full system's balanced performance across strategic metrics, confirming that oversight and context management are *complementary capabilities* that work synergistically for effective long-horizon reasoning.

## 5 Practical Extensions

In addition to the core framework, we introduce two extensions that further enhance robustness and efficiency: (1) a specialized compact context manager, **Context-12B**, trained via supervised fine-tuning (SFT) and direct preference optimization (DPO) to reduce token cost while maintaining strong performance , and (2) a test-time scaling variant, **COMPASS-TTS**, which leverages parallel sampling to improve reliability under uncertainty.

### 5.1 Context-12B: Training Specialized Context Managers

While larger models excel at summarization and context organization, their high API costs and deployment overhead hurts the efficiency of the system. Our case analyses as shown in Appendix §E.6 further revealed that, among the three COMPASS agents, the *Context Manager* is the most structured and deterministic, operating more like a summarizer than an open-ended reasoner. This observa-

Table 3: Ablation studies for the Meta-Thinking and Context Management components on **BrowseComp**. The full system is presented as a reference. Each subsequent section ablates one component, showing a marked drop in success and strategic metrics compared to the full system. **Strategy Adequacy** represents the average of 4 metrics.

| Component Configuration | Pass @ 1 (%) | Component Tokens* | Total Tokens | PAR | PVR | CA | ERC | Strategy Adequacy |
|---|---|---|---|---|---|---|---|---|
| ***Reference: Full System*** | | | | | | | | |
| Gemini 2.5 Pro (Full System) | **35.4** | 59K | 185K | 0.85 | **0.48** | 0.88 | **0.55** | **0.69** |
| ***Ablation 1: Meta-Thinking Agent*** *(Context Manager/Main Agent fixed to Gemini 2.5 Pro)* | | | | | | | | |
| None (Main + Context only) | 15.2 | 0K | 85K | **0.92** | 0.12 | **0.95** | 0.21 | 0.55 |
| Gemini 2.5 Flash | 28.5 | 8K | 132K | 0.79 | 0.33 | 0.78 | 0.38 | 0.57 |
| Gemini 2.5 Pro w/ Extra Tools | 32.8 | 14K | 140K | 0.82 | 0.41 | 0.76 | 0.45 | 0.61 |
| ***Ablation 2: Context Manager*** *(Meta-Thinking/Main Agent fixed to Gemini 2.5 Pro)* | | | | | | | | |
| None (No Context Manager) | 26.4 | 0K | 156K | 0.91 | 0.18 | 0.94 | 0.25 | 0.57 |
| Gemma-3-12B | 28.5 | 12K | 144K | 0.77 | 0.35 | 0.76 | 0.44 | 0.58 |
| Gemini 2.5 Flash | 31.8 | 20K | 212K | 0.74 | 0.52 | 0.80 | 0.54 | 0.65 |

*Component Tokens measure usage for the specific agent being ablated (Meta-Thinking or Context Manager). For the Total, it's the sum of all three agents.

tion motivates us to design a smaller, deployable model that retains the research-status-oriented summarization ability of larger models without their computational footprint.

**Data Collection.** We leverage **Gemini 2.5 Pro** with searching tool as a *data engine* to generate high-quality training signals. Using COMPASS rollouts on Knowledge-intense long-horizon benchmarks or complex academic QA (GAIA, SimpleQA, MMLU-Pro, etc), we extract training data where the input is the full reasoning trajectory plus meta-thinking notes, and the output is the optimized brief provided to the Main Agent. To ensure quality, we apply filtering: (i) remove trivial trajectories with fewer than three tool interactions, (ii) exclude degenerate completions where the correct answer is reached without reflection, and (iii) upsample cases where context management clearly drives recovery or proper task termination.

**Training Pipeline.** We first distill this capability into **Gemma-3 12B**, obtaining **Context-12B-SFT**. Supervised fine-tuning teaches the model to follow the instructions and produce concise, strategically aligned briefs from complex trajectories from bigger models with our collected data. We then refine Context-12B-SFT using **direct preference optimization (DPO)**. For each training trajectory, we sample multiple candidate summaries and construct preference pairs from the same data we used for SFT: We continue applying the generated contexts in the COMPASS inference engine, and selecting the context leading successful completions with fewer tokens are labeled as preferred, while redundant or error-prone summaries are rejected.(more training details can be referred in Appendix §C.4) This produces the final model, **Context-12B**, optimized for both accuracy and efficiency.

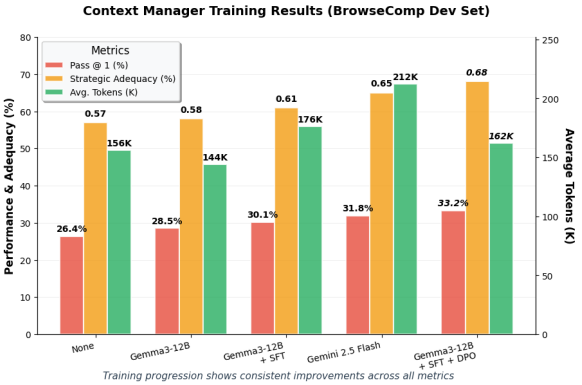


Figure 4: Context-12B Performance on BrowseComp. Pass @ 1(%), strateg adequacy (×100), and token efficiency all improve progressively. DPO yields substantial efficiency gains without sacrificing accuracy.

**Results.** As shown in Figure 4, we evaluated the performance of Context-12B on Browsecomp, achieving performance comparable to larger models (Gemini 2.5 Flash) while using only 70% of their tokens with a SFT-DPO training pipeline .

## 5.2 COMPASS-TTS: Test-time Scaling with Parallel Sampling

Parallel sampling during inference is a common strategy to improve the performance (Wang et al., 2023b; Han et al., 2025). With $n$ as parallel samples, we extend the framework with **COMPASS-TTS**, which explores multiple reasoning or context-management alternatives concurrently as (1) **Full-pipeline sampling (Full-PS).** Executes $n$ diversified runs of the *entire pipeline* (varying seeds and temp). A lightweight synthesizer $g$ then aggregates the candidate outputs into a single final answer. (2) **Meta-thinking sampling (MT-PS).** Parallelizes only the meta-thinking module, producing $n$ alternative triggering and decision pro-

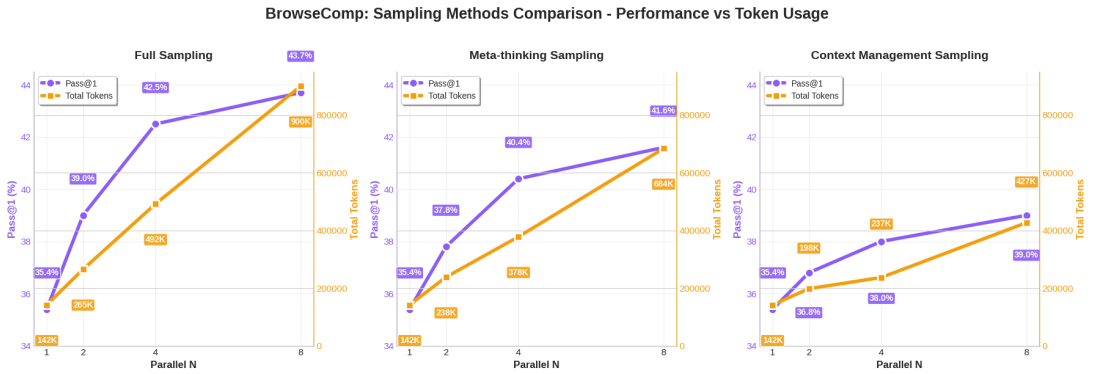


Figure 5: Performance (Pass@1) vs. token cost for three COMPASS-TTS sampling methods on the BrowseComp benchmark. Increasing the number of parallel samples improves accuracy but also raises token costs.

posals. A synthesizer with slightly different prompt than the Full-PS's one merges these into one coherent downstream plan, which is then executed once. (3) **Context-management sampling (CM-PS).** Parallelizes only the context manager, yielding $n$ alternative contextualization. These are distilled by the synthesizer into a single injected context before task execution.

**Results.** Figure 5 plots accuracy and total tokens for $n \in \{1, 2, 4, 8\}$. All methods improve monotonically over $n = 1$, with Full-PS showing the strongest gains but steepest token growth, MT-PS offering a balanced middle ground, and CM-PS achieving the best efficiency. Performance plateaus around $n = 4$, suggesting that $n = 2$–$4$ offers a practical sweet spot to balance accuracy and costs.

## 6 Related Work

**LLM-based Agentic Systems.** LLM-based agents primarily follow two paradigms. **Single-agent systems** extend the ReAct-style think–act–observe loop (Yao et al., 2023; Shinn et al., 2023) into multi-turn settings. They are popular for their simplicity and autonomy, and have been enhanced via reinforcement learning (Zhang et al., 2025) to yield post-trained, tool-integrated models (Comanici, 2025; Li et al., 2025b; Feng et al., 2026; Jin et al., 2025) and inference-time reasoning and memory modules (Anthropic, 2025; Wang et al., 2024). **Multi-agent systems (MAS)** instead distribute reasoning across specialized roles for improved robustness (Liang et al., 2024; Wu et al., 2024; Tran et al., 2025). MAS employ explicit coordination by centralized managers, decentralized handoffs, or predefined workflows (Anthropic, 2024) and often achieve state-of-the-art results on complex benchmarks (Huang et al., 2025; Wei et al., 2025; Snell et al., 2025; Li et al., 2023). However, their reliance on manually designed pipelines limits scalability and generalization (Sapkota et al., 2025; Pan et al., 2025). Our work instead makes *strategic reasoning* and *context management* explicit architectural components while retaining the fluid, end-to-end nature of single-agent systems.

**LLM Reasoning for Long-Horizon Tasks.** While Chain-of-Thought (CoT) reasoning improves short reasoning tasks (Wei et al., 2022), it falters on long-horizon problems where errors accumulate over multiple steps (Chen et al., 2025; Zhang et al., 2023). Recent efforts enhance long-horizon reasoning through hierarchical control (Wang et al., 2025; Chen et al., 2025) and explicit planning or self-reflection, using prompting (Sun et al., 2023; Madaan et al., 2023), post-training (et al., 2025; Parmar et al., 2025), or both (Erdogan et al., 2025). Yet, these methods often remain brittle or superficial (Liu, 2025; Lindsey, 2025; Huang et al., 2024a), especially under incomplete or excessive context. Insights from *context engineering* studies highlight how redundant inputs can degrade reasoning quality (Hong et al., 2025; Mei et al., 2025; Li et al., 2025c; Liu et al., 2024). Motivated by this, our framework integrates adaptive context selection into the reasoning process to mitigate cognitive load and improve adaptivity under long horizon.

## 7 Conclusion

We presented **COMPASS**, a hierarchical framework that elevates strategic reasoning and context management to *architectural primitives*, enabling reliable long-horizon reasoning without the complexity of multi-agent topologies. Our findings highlight the critical distinction between *tactical reasoning*—the capacity for accurate tool use and instruction following within an agent turn—and *strategic reasoning*—the higher-level oversight that

guides reflection, replanning, and termination decisions. We further highlight the role of *context management* in maintaining coherence and adaptability across extended trajectories, offering actionable principles for developing scalable and robust agentic systems in long-horizon reasoning tasks.

## Limitations

While COMPASS demonstrates strong performance on QA-style agentic benchmarks, our evaluation currently emphasizes controlled reasoning environments to clearly illustrate the framework's proof of concept. Specifically, experiments are conducted within settings that involve searching, text-based browsing, and code execution tools. Future extensions to more open-ended domains—and the integration of richer interoperability mechanisms such as MCP servers and agent-to-agent (A2A) communication protocols (Hou et al., 2025)—would enable a more comprehensive assessment of COMPASS's robustness in dynamic, real-world contexts. In addition, Our study primarily focuses on proprietary frontier models, as our goal is to incentivize and analyze the limits of state-of-the-art reasoning capabilities on challenging long-horizon tasks. Common open-source models currently underperform significantly on such tasks (Plaat, 2025), but future work can extend our framework to investigate their performance and post-training potential.

## Acknowledgements

We thank Google for providing access to Gemini 2.5 Pro/Flash and Gemma-3 via Google Cloud Vertex AI, whose infrastructure supported all experiments and training runs in this work. G. Wan and S. Li are supported by the U.S. Office of Naval Research Award under Grant Number N00014-24-1-2668, the National Science Foundation under Grants IIS-2316306 and CNS-2330215, and the National Institutes of Health (NIH) under Grant R01EB293388.

## References

Anthropic. 2024. Building effective agents. https://www.anthropic.com/engineering/building-effective-agents. Accessed: 2025-09-24.

Anthropic. 2025. The "think" tool: Enabling claude to stop and think in complex tool use situations. https://www.anthropic.com/engineering/claude-think-tool. Accessed: 2025-09-22.

Trishna Chakraborty, Udita Ghosh, Xiaopan Zhang, Fahim Faisal Niloy, Yue Dong, Jiachen Li, Amit K. Roy-Chowdhury, and Chengyu Song. 2025. Heal: An empirical study on hallucinations in embodied agents driven by large language models. *Preprint*, arXiv:2506.15065.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *Preprint*, arXiv:2503.09567.

Gheorghe et al. Comanici. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending llm context window beyond 2 million tokens. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*.

Lutfi Eren Erdogan, Hiroki Furuta, Sehoon Kim, Nicholas Lee, Suhong Moon, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. 2025. Plan-and-act: Improving planning of agents for long-horizon tasks. In *Forty-second International Conference on Machine Learning*.

DeepSeek-AI et al. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Preprint*, arXiv:2501.12948.

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

In Gim, Guojun Chen, Seung seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. 2024. Prompt cache: Modular attention reuse for low-latency inference. *Preprint*, arXiv:2311.04934.

Gonzalo Gonzalez-Pumariega, Leong Su Yean, Neha Sunkara, and Sanjiban Choudhury. 2025. Robotouille: An asynchronous planning benchmark for llm agents. *Preprint*, arXiv:2502.05227.

Rujun Han, Yanfei Chen, Zoey CuiZhu, Lesly Miculicich, Guan Sun, Yuanjun Bi, Weiming Wen, Hui Wan, Chunfeng Wen, Solène Maître, George Lee, Vishy Tirumalashetty, Emily Xue, Zizhao Zhang, Salem Haykal, Burak Gokturk, Tomas Pfister, and Chen-Yu Lee. 2025. Deep researcher with test-time diffusion. *Preprint*, arXiv:2507.16075.

Kelly Hong, Anton Troynikov, and Jeff Huber. 2025. Context rot: How increasing input tokens impacts llm performance. Technical report, Chroma Research. Technical report.

Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. 2025. Model context protocol (mcp): Landscape, security threats, and future research directions. *Preprint*, arXiv:2503.23278.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024a. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024b. Understanding the planning of llm agents: A survey. *Preprint*, arXiv:2402.02716.

Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Huichi Zhou, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, Kun Shao, and Jun Wang. 2025. Deep research agents: A systematic examination and roadmap. *Preprint*, arXiv:2506.18096.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Hang Li, Yucheng Chu, Kaiqi Yang, Yasemin Copur-Gencturk, and Jiliang Tang. 2025a. Llm-based automated grading with human-in-the-loop. *Preprint*, arXiv:2504.05239.

Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, Hongxuan Lu, Tianrui Qin, Chenghao Zhu, Yi Yao, Shuying Fan, Xiaowan Li, Tiannan Wang, Pai Liu, King Zhu, and 11 others. 2025b. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl. *Preprint*, arXiv:2508.13167.

Xiaomin Li, Zhou Yu, Zhiwei Zhang, Xupeng Chen, Ziji Zhang, Yingying Zhuang, Narayanan Sadagopan, and Anurag Beniwal. 2025c. When thinking fails: The pitfalls of reasoning for instruction-following in llms. *Preprint*, arXiv:2505.11423.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA. Association for Computational Linguistics.

Jack et al. Lindsey. 2025. On the biology of a large language model. `https://transformer-circuits.pub/2025/attribution-graphs/biology.html`. Transformer Circuits; published March 27, 2025; accessed 2025-09-10.

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

Zichen et al. Liu. 2025. There may not be aha moment in r1-zero-like training — a pilot study. `https://oatllm.notion.site/oat-zero`. Notion blog; accessed 2025-09-10.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, Chenlin Zhou, Jiayi Mao, Tianze Xia, Jiafeng Guo, and Shenghua Liu. 2025. A survey of context engineering for large language models. *Preprint*, arXiv:2507.13334.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. GAIA: a benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations*.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2025. Large language models: A survey. *Preprint*, arXiv:2402.06196.

Melissa Z Pan, Mert Cemri, Lakshya A Agrawal, Shuyi Yang, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Kannan Ramchandran, Dan Klein, Joseph E. Gonzalez, Matei Zaharia, and Ion Stoica. 2025. Why do multiagent systems fail? In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

Mihir Parmar, Palash Goyal, Xin Liu, Yiwen Song, Mingyang Ling, Chitta Baral, Hamid Palangi, and Tomas Pfister. 2025. Plan-tuning: Post-training language models to learn step-by-step planning for complex problem solving. *Preprint*, arXiv:2507.07495.

Long et al. Phan. 2025. Humanity's Last Exam. *Preprint*, arXiv:2501.14249.

Aske et al. Plaat. 2025. Survey on agentic large language models: reasoning, acting, and interacting. *ACM Computing Surveys*.

Rafael et al. Rafailov. 2024. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Ranjan Sapkota, Konstantinos I. Roumeliotis, and Manoj Karkee. 2025. Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges. *Information Fusion*, 126:103599.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*.

Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. 2023. Adaplanner: Adaptive planning from feedback with language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wannita Takerngsaksiri, Jirat Pasuksmit, Patanamon Thongtanunam, Chakkrit Tantithamthavorn, Ruixiong Zhang, Fan Jiang, Jing Li, Evan Cook, Kun Chen, and Ming Wu. 2025. Human-in-the-loop software development agents. In *2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 342–352.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of llms. *Preprint*, arXiv:2501.06322.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*.

Haozhe Wang, Qixin Xu, Che Liu, Junhong Wu, Fangzhen Lin, and Wenhu Chen. 2025. Emergent hierarchical reasoning in llms through reinforcement learning. *Preprint*, arXiv:2509.03646.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *Preprint*, arXiv:2504.12516.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, Wei He, Yiwen Ding, Guanyu Li, Zehui Chen, Zhengyin Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Tao Gui, and 4 others. 2025. Agentgym-rl: Training llm agents for long-horizon decision making through multi-turn reinforcement learning. *Preprint*, arXiv:2509.08755.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2025. Hallucination is inevitable: An innate limitation of large language models. *Preprint*, arXiv:2401.11817.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Yue Liao, Hongru Wang, Mengyue Yang, Heng Ji, and 5 others. 2025. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2023. How language model hallucinations can snowball. *Preprint*, arXiv:2305.13534.

# A Reproducibility

To ensure reproducibility of our results, we provide comprehensive implementation details and experimental specifications throughout this work. Section 4 and Appendix C detail our experimental setup, including specific model versions (Gemini 2.5 Pro/Flash), hyperparameters, and evaluation metrics. All prompt templates for the three COMPASS agents are provided in Appendix B, along with evaluation prompts for our strategic reasoning metrics. Our training pipeline for Context-12B is fully specified in Appendix C.4, including data preprocessing steps, SFT and DPO hyperparameters, and infrastructure requirements. Detailed ablation studies in Table 3 isolate the contribution of each component. Our evaluation covers three established benchmarks (GAIA, BrowseComp, HLE) with standard Pass@1 metrics, supplemented by novel strategic reasoning metrics that are operationally defined with LLM-as-a-Judge protocols detailed in Appendix B. Case studies in Appendix E provide concrete trajectory examples illustrating key failure modes and recovery patterns. All experiments used Google Cloud infrastructure with specific GPU configurations and API endpoints documented in Appendix C. While our implementation relies on proprietary Gemini models, the architectural principles and algorithmic framework are model-agnostic and can be adapted to other LLM backends.

# B Prompt Templates

We report here the full prompt templates used in evaluation and in the definition of **COMPASS**. The evaluation templates are used for accuracy and strategic reasoning metrics, while the agent templates specify the architectural roles of COMPASS's components.

## B.1 Evaluation Prompts

**Accuracy Evaluation.** We evaluate correctness (Pass@1) using a query template that enforces a canonical answer format, and a grader template that judges against the gold label.

**Query Template**

```
{Question}
Your response should be in the following format:
Explanation: {{your explanation for your final answer}}
Exact Answer: {{your succinct, final answer}}
```

**Grader Template**

```
Judge whether the following [response] to [question] is correct or not based on the precise and unambiguous [correct_answer] below.
[question]: {question}
[response]: {response}
[correct_answer]: {correct_answer}
Your judgement must be in the format and criteria specified below:
extracted_final_answer: The final exact answer extracted from the [response]. Put 'None' if there is no exact, final answer.
reasoning: Explain why the extracted_final_answer is correct or incorrect based only on [correct_answer].
correct: Answer 'yes' if extracted_final_answer matches [correct_answer] (with small tolerance for numerical error), otherwise 'no'.
```

**Strategic Reasoning Metrics.** The following prompts evaluate the four strategic reasoning metrics (PAR, PVR, CA, ERC) defined in Section 2. Each template requires both a justification and a binary decision.

**Persist Appropriateness Rate (PAR)**

```
[state]: {trajectory_state}
[decision]: {decision}
reasoning: Explain whether persisting was appropriate in this situation.
par_correct: Answer 'yes' if persisting was the right choice, otherwise 'no'.
```

**Pivot Recognition (PVR)**

```
[state]: {trajectory_state}
[decision]: {decision}
reasoning: Explain whether pivoting was the right action here.
pvr_correct: Answer 'yes' if pivoting was appropriate, otherwise 'no'.
```

**Conclude Accuracy (CA)**

```
[state]: {trajectory_state}
[decision]: {decision}
[ground_truth_answer]: {correct_answer}
reasoning: Explain whether concluding was accurate given the ground truth.
ca_correct: Answer 'yes' if conclusion matches the correct answer, otherwise 'no'.
```

**Error-Recovery Continuation (ERC)**

```
[state]: {trajectory_state}
[decision]: {decision}
[ground_truth_answer]: {correct_answer}
```

**reasoning:** Explain whether continuing was necessary for error recovery.
**erc_correct:** Answer 'yes' if continuation was the correct action to avoid submitting an incorrect answer, otherwise 'no'.

### B.2 COMPASS Agent Prompts

We include here the high-level instructions provided to each component in COMPASS. These prompts implement the architectural separation introduced in Section 2: a Main Agent for tactical execution, a Context Manager for maintaining and synthesizing context, a Meta-Thinker for asynchronous strategic oversight, and an Answer Synthesizer for producing the final output. (Note that these prompts are for demonstration purpose, and the exact prompts should adapt to different benchmarks as needed).

**Main Agent Prompt**

You are the Main Agent. Your role is to execute the user's task through an iterative loop of reasoning, tool use, and observation. At each step you must:

1. Read the current task context provided to you.
2. Decide on one action or tool call to perform.
3. Execute only one tool at a time (e.g., search, retrieve, write, verify).
4. Observe the result and update your reasoning.
5. Repeat until you believe the task is complete.

**Guidelines:**

- Focus on tactical execution using the immediate, curated context provided.
- Do not attempt to monitor or evaluate global progress yourself.
- Be explicit in reasoning: explain why the chosen tool is relevant.
- Stop execution and return your final answer when you believe the task requirements are satisfied.

**Meta-Thinker Prompt**

You are the Meta-Thinker. You run asynchronously in parallel to the Main Agent. Your job is to monitor execution for strategic anomalies and completion signals.
**Your tasks:**

1. Continuously observe the Main Agent's actions and outputs.
2. Detect anomalies such as repeated failures, contradictions, or wasted effort.
3. Detect signals that suggest the task may be complete.
4. When triggered, decide whether to:
   - Persist (allow the Main Agent to continue).
   - Pivot (redirect strategy).
   - Verify (pause and request additional checks).
   - Terminate (stop execution and return the final answer).

**Guidelines:**

- Your monitoring is lightweight and only activates when necessary.
- Issue intervention or stopping signals explicitly when you detect risk of compounding errors or when the answer is already sufficient.
- Do not duplicate the Main Agent's work; focus on higher-level judgment.

**Context Manager Prompt**

**Role.** You are the **Context Manager**. You transform full task history and research notes into a concise, execution-ready context for the Main Agent each turn.
**Inputs (provided to you):**

- **Original Question** $q$.
- **Previous Notes** $n_t$: structured briefs from earlier rounds (high-level summaries, verified evidence, constraints).
- **Current Research Progress** $\mathcal{T}_t$: the turn-level trace (key observations, tool results, reflections).
- **Meta Decision** decision: continue/reflect/replan/verify/stop.

**Your Single Output:** produce a structured context $x_t$ for the Main Agent (concise, specific, and directly actionable). Use the following sections:

1. **Task**: one-sentence restatement of $q$.
2. **Most-Recent Evidence**: 2–5 bullet points of verified, relevant facts (cite tool names/sources inline, if applicable).
3. **Critical Constraints & Corrections**: formatting/grounding constraints (e.g., "must cite FDA source"), and any corrections to earlier mistakes.
4. **Open Items**: unresolved sub-questions or missing data (prioritized).

5. **Next Actions (Plan)**: 2–4 concrete steps aligned with decision (tools to call, targets, success criteria).

6. **Tool Hints (Optional)**: specific tools to use

**Guidelines:**

- Be strictly selective: include only information required for the next turn's tactical reasoning.
- Do not execute tasks; do not duplicate raw history—promote only salient facts and decisions.
- Keep the output compact (typically $\leq$ 200–300 tokens), with bullet lists over prose when possible.

### Answer Synthesizer Prompt

**Role.** You are the **Answer Synthesizer**. When the Meta-Thinker signals completion, you generate the final user-facing answer.
**Inputs:**

- Final trace (key observations and reasoning).
- Notes $n_t$ (structured briefs from prior rounds).
- Original question $q$ and any explicit constraints.

**Your Tasks:**

1. Integrate evidence from $\mathcal{T}_t$ and $n_t$.
2. Prioritize verified, high-confidence findings and resolve minor inconsistencies.
3. Produce a clear, direct answer to $q$.

**Guidelines:**

- Use concise, authoritative natural language.
- If needed, include a one-line justification citing key evidence sources.
- Avoid hedging such as "insufficient information"; provide your strongest synthesis.

### Note Append (Pseudocode)

```
# Inputs:
#   turn_id      : integer round index t
#   context_text : structured output from
     Context Manager (Sections 1-6)
# Output:
#   note store updated with sections 2-4 for
     round t

function AppendNote(turn_id, context_text):
    # 1) Parse only the Context Manager
         sections we keep as notes:
    #    2) Most-Recent Evidence
    #    3) Critical Constraints & Corrections
    #    4) Open Items
    evidence   <- ExtractBullets(context_text,
          header="Most-Recent Evidence")
    constraints <- ExtractBullets(context_text
         , header="Critical Constraints &
         Corrections")
    open_items <- ExtractBullets(context_text,
          header="Open Items")

    # 2) Form a compact record tagged by round
         :
    record <- {
        round: turn_id,
        evidence: evidence,
        constraints: constraints,
        open_items: open_items
    }

    # 3) Append to the rolling note store and
         persist:
    NoteStore.Append(record)
    NoteStore.Save()       # implementation-
         specific persistence (e.g., InMemory)

# Helper: returns a list of bullet lines
     under a given section header.
function ExtractBullets(text, header):
    # Locate 'header:' line (with optional
         numbering like "2) header:")
    # Collect subsequent bullet/numbered lines
          until next top-level header.
    bullets <- []
    ...      # implementation-specific parsing
    return bullets
```

## C Resources, Data, and Training

Here is the inner loop, Re-act style algorithm:

### C.1 Benchmarks

We evaluate COMPASS on benchmarks that stress long-horizon reasoning across multiple interactions. Short-form QA datasets such as **SimpleQA** and **GPQA** were piloted but omitted, since strong base models already achieve near-saturation and these tasks do not benefit from tool augmentation. Our main evaluation therefore focuses on **DeepResearch-relevant benchmarks** that typically require more than 20 reasoning–action steps:

- **GAIA** (Mialon et al., 2024): all Level 1–3 tasks without images, spanning diverse scientific and commonsense domains.
- **BrowseComp** (Wei et al., 2025): 1,266 questions requiring sustained web navigation,

**Algorithm 2** ExecuteTurn: Inner Loop of Tactical Reasoning.

**Require:** $\mathcal{A}^{\text{main}}$, $\mathcal{A}^{\text{meta}}$, *context* $x$, max inner iterations $I_{\max}$
**Ensure:** Partial trace $\mathcal{T}$

$\mathcal{T} \leftarrow []$
$\mathcal{A}^{\text{meta}}$.StartMonitoring($\mathcal{T}$) ▷ Begin async monitoring of trace queue
**for** $i = 0, 1, \ldots, I_{\max} - 1$ **do**
  **if** $\mathcal{A}^{\text{meta}}$.IsTriggered() **or** $\mathcal{A}^{\text{main}}$.HasFinalAnswer() **then**
    **break** ▷ Exit on anomaly or completion
  **end if**
  *step* $\leftarrow \mathcal{A}^{\text{main}}$.Execute($x, \mathcal{T}, \mathcal{O}$)
  $\mathcal{T}$.append(*step*) ▷ Meta-agent observes queue changes asynchronously
**end for**
$\mathcal{A}^{\text{meta}}$.StopMonitoring() ▷ End async monitoring
**return** $\mathcal{T}$

cross-source verification, and entangled fact retrieval with short, verifiable answers.

- **Humans' Last Exam (HLE)** (Phan, 2025): 2,158 questions across mathematics, humanities, and natural sciences, excluding image-based or non-tool-relevant cases.

These benchmarks collectively cover diverse failure modes in long-horizon reasoning: cascading search errors, tool API misuses, and premature stopping.

### C.2 Baselines and Models

We compare against two categories. **Fundamental paradigms**: single-agent (*Search/Browse*, *+Meta-Thinking*, *+Context*) and multi-agent (*Agent-as-a-Tool*, *Decentralized Handoffs*, *Plan-and-Execute* (Wang et al., 2023a)). **Established research agents**: OpenAI *DeepResearch*, DeepSeek Agent, and Google *TestTime Diffusion*. For fair comparison, test-time scaling (parallel sampling) is included only when benchmarking against established systems. All experiments use **Gemini 2.5 Pro** and **Gemini 2.5 Flash** as backbone reasoning models, with **Gemma-3-12B** in ablations for specialized Context Manager training. For reproducibility, we fix random seeds across sampling runs and log all trajectories for post-hoc auditing.

### C.3 Evaluation Metrics

Our primary metric is **accuracy (Pass@1)** defined per benchmark-specific criteria. We additionally measure **token usage** (both per-step and end-to-end) to capture efficiency. To assess strategic reliability, we introduce four **meta-thinking metrics** (PAR, PVR, CA, ERC) as formalized in Section 2, each judged via LLM-as-a-Judge with structured outputs. See Appendix B for full grading prompts. We report mean values over three runs to mitigate stochasticity from sampling and tool call variability.

### C.4 Synthetic Data and Training for Context Manager

We construct training data by mining trajectories from GAIA, SimpleQA, HotPotQA, and GPQA, MMLU or MMLU-Pro, in order to cover different reasoning scenarios, using the COMPASS inference pipeline. From 13,486 raw trajectories, we filter down to **2,065** high-quality examples emphasizing (i) complete task solutions, (ii) error-recovery sequences, and (iii) proper termination after intermediate success. We exclude trivial paths ($< 3$ tool calls) and degenerate answers without reflection, and we upsample recovery cases to improve robustness. Each example pairs trajectory history and Meta-Thinker reflections with the optimized context for the next turn.

#### C.4.1 Data Preprocessing and Quality Control

Our data construction pipeline applies several quality filters to ensure training examples capture the structured reasoning patterns required for context management:

**Trajectory Length Filtering:** We retain only trajectories with 3-25 tool calls, excluding both trivial single-step solutions and excessively long sequences that may contain repetitive failures.

**Success Pattern Analysis:** Training examples are categorized into three types: (i) *direct success* trajectories that solve tasks without errors, (ii) *recovery sequences* where agents overcome initial mistakes through strategic pivots, and (iii) *verification patterns* where agents validate solutions before concluding. We upsample recovery sequences (2.3× multiplier) to improve the model's ability to synthesize context after failures.

**Context Complexity Stratification:** We balance examples across varying context complexities: simple constraint tracking (35%), multi-source evidence synthesis (45%), and complex constraint

interaction cases (20%). This ensures the model learns both basic summarization and sophisticated context organization.

#### C.4.2 Supervised Fine-Tuning (SFT)

We fine-tune the model to generate concise context briefs that preserve critical constraints and strategic signals from Meta-Thinker reflections while maintaining structured output format for pipeline integration.

**Dataset Construction:** We curate a training dataset of 10,347 examples comprising:

- Query-context pairs where contexts distill verbose planning queries into minimal briefs while preserving all critical constraints
- Contexts derived from Meta-Thinker reflections, maintaining strategic planning signals
- Structured outputs following our standardized template (objective restatement, progress tracking, constraint preservation, next-step context)

The dataset is constructed from COMPASS pipeline executions, with human annotation to ensure quality and template adherence.

**Training Objective:** We use standard cross-entropy loss for autoregressive language modeling:

$$\mathcal{L}_{SFT} = -\sum_{t=1}^{T} \log P(y_t \mid y_{<t}, x; \theta)$$

where $x$ represents the input query with Meta-Thinker reflections, and $y$ is the target context brief.

**Training Setup:** We fine-tune **Gemma-3-12B** for 3,000 steps on four A100 GPUs (80GB) using DeepSpeed ZeRO Stage-2 with gradient accumulation. Hyperparameters: AdamW optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$), effective batch size 32, learning rate $1 \times 10^{-4}$ with cosine decay to $1 \times 10^{-6}$, weight decay 0.05, warmup ratio 0.1. We apply gradient clipping (max norm 1.0) and monitor validation perplexity every 200 steps, selecting the checkpoint with lowest perplexity on a held-out validation set ($N = 500$).

**Template Consistency:** All training examples follow our structured format, ensuring downstream compatibility with the COMPASS pipeline. We validate format compliance through post-processing checks, achieving 94.3% adherence on held-out test data.

#### C.4.3 Direct Preference Optimization (DPO)

Following similar procedure as the SFT above, we apply DPO (Rafailov, 2024) to further align the context generation model toward task-efficient summaries. DPO enables direct optimization from preference data without requiring explicit reward models or reinforcement learning.

**Preference Data Collection:** For each of training queries from the above SFT dataset, we generate four candidate context summaries from Context-12B-SFT using nucleus sampling with varying temperatures $\tau \in \{0.7, 0.9, 1.1, 1.3\}$ and top-$p = 0.95$. Each candidate is evaluated by executing the full COMPASS pipeline and measuring:

- Task success (binary completion indicator)
- Token efficiency (total tokens consumed until completion or timeout)
- Strategic appropriateness (composite score from PAR/PVR/CA/ERC metrics)

**Preference Pair Construction:** We rank the four candidates per trajectory by a composite metric: score $= \mathbf{1}_{\{\text{success}\}} - 0.001\,\text{tokens}$ , where the token penalty encourages efficiency without sacrificing success. We construct preference pairs $(y_w, y_l)$ from adjacent rankings (rank $i$ vs. rank $i+1$), yielding up to 6 pairs per trajectory.

To ensure meaningful preferences, we filter pairs where: (1) both contexts lead to failure, (2) the success rate difference is $< 5\%$ on our validation set, or (3) the preferred context uses $> 20\%$ more tokens without measurable quality improvement. This filtering yields 8,200 high-quality preference pairs.

**Training Setup:** We optimize the DPO objective:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x,y_w,y_l)}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right]$$

where $\pi_{\text{ref}}$ is Context-12B-SFT (frozen), $\pi_\theta$ is the policy being optimized, and $\beta = 0.1$ controls the KL penalty strength.

Training runs for 6,000 steps on eight A100 GPUs with DeepSpeed ZeRO Stage-2. Hyperparameters: AdamW optimizer, batch size 32 (4 per GPU), learning rate $5 \times 10^{-5}$ with linear warmup (10% of steps) and cosine decay, gradient clipping

(max norm 1.0). We evaluate on a held-out validation set of 200 trajectories every 500 steps and select the checkpoint maximizing validation success rate.

**Results:** The final DPO model (Context-12B-DPO) achieves 30% token reduction compared to Context-12B-SFT (from 2,847 to 1,993 average tokens per task) while maintaining comparable task success rates (87.3% vs. 87.8% on our test set). Notably, the DPO model shows improved strategic appropriateness scores, suggesting better alignment with efficient planning trajectories.

### C.5 Resources and Infrastructure

All Gemini API calls and long-term memory operations were executed through services and infrastructure provide by google cloud infrastructure, leveraging its managed memory bank services for trajectory storage and retrieval. Training was performed with up to 8 A100-80GB GPUs and huggingface's trl library was used for SFT and DPO implementations. We use Agents Development Kit (ADK) library to implement multi-agent orchestration. To ensure reproducibility, we log tool API events, token counts, and intermediate contexts; all experiments were run with fixed seeds and capped API retries to handle stochastic tool failures.

**Computational Requirements:** Total training compute for Context-12B required approximately 480 GPU-hours across SFT and DPO phases. Data preprocessing and preference pair construction consumed an additional 120 CPU-hours. All training runs used mixed precision (fp16) with gradient checkpointing to manage memory usage efficiently.

## D Cost Efficiency via Context Compaction and Prompt Caching

The asynchronous, turn-level design of COMPASS is particularly amenable to prompt caching. To illustrate why, consider a simplified comparison between a standard iterative-refinement trajectory and COMPASS on a long-horizon task.

**Iterative-Refinement Trajectory.** In a standard iterative-refinement setup, the agent revises its active plan or scratchpad after each unsuccessful attempt. This prevents the prompt from growing indefinitely, but it also means that the prompt prefix is frequently *rewritten* rather than simply extended. Under prefix-based prompt caching, such updates reduce reuse across successive calls:

```
Round 1 input: [System Prompt | Query |
Plan_1 | Attempt_1]
Round 2 input: [System Prompt | Query |
Plan_2 | Attempt_2]
Round 3 input: [System Prompt | Query |
Plan_3 | Attempt_3]
. . .
```

Because the active plan is updated across rounds, the shared prefix between successive calls is often limited to the static system prompt and query, leading to relatively low cache reuse. In our experiments, this baseline yields a cache hit rate of roughly $\sim$9%.

**COMPASS Trajectory.** In COMPASS, the Context Manager produces a *structured brief* at the end of each outer-loop iteration, which becomes the fixed prefix for all inner-loop steps in the next iteration. The Main Agent then executes multiple inner steps conditioned on the same refreshed context $x_t$, so that this compact brief is reused as a stable cached prefix across all calls within that iteration:

```
- Outer iteration t -
Context brief x_t: [System Prompt | Query
| Brief_t]                          cached


Inner step 1 input: [System Prompt |
Query | Brief_t]
Inner step 2 input: [System Prompt |
Query | Brief_t
            | Think_1 | Act_1 | Obs_1]
Inner step 3 input: [System Prompt |
Query | Brief_t
            | Think_1 | Act_1 | Obs_1
            | Think_2 | Act_2 | Obs_2]


- Outer iteration t+1 -
Context brief x_{t+1}: [System Prompt |
Query | Brief_{t+1}]                cached


Inner step 1 input: [System Prompt |
Query | Brief_{t+1}]
Inner step 2 input: [System Prompt |
Query | Brief_{t+1}
            | Think_1 | Act_1 | Obs_1]
. . .
```

The key difference is that COMPASS refreshes context only at outer-loop boundaries. Within each outer iteration, the system prompt, query, and structured brief form a stable shared prefix that is reused across all inner-loop steps, while only the incremental thought-action-observation triples introduce new tokens. This design combines *context compaction*, which prevents unbounded prompt growth, with *prompt caching*, which increases reuse of repeated prefixes across calls. In our experiments,

COMPASS achieves a substantially higher cache hit rate (∼41%) than the iterative- refinement baseline (∼9%), helping offset the cost of multi-agent orchestration.

**Metric.** We measure cache hit rate as the fraction of total input tokens across all model calls that were served from the provider's prompt cache.

## E Case Studies

This section provides detailed trajectory examples illustrating the meta-thinking decision scenarios outlined in Table 1. We use the following complex query from BrowseComp to demonstrate failure patterns in long-horizon tasks:

**Query:** Between 1990 and 1994 inclusive, what teams played in a soccer match with a Brazilian referee had four yellow cards, two for each team where three of the total four were not issued during the first half, and four substitutions, one of which was for an injury in the first 25 minutes of the match.

**Ground Truth Answer:** Ireland v Romania

### E.1 Scenario 1A: Local Error - Correct Persistence

This trajectory demonstrates appropriate persistence when facing a local search refinement need.

```
Step 1: <think>
Need to find soccer match 1990-1994 with
    specific yellow card and substitution
    patterns with Brazilian referee.
</think>
<tool_call name="search">
soccer match 1990-1994 Brazilian referee
    yellow cards substitutions
</tool_call>
<tool_response>
Results show various World Cup matches from
    1990-1994, but no specific yellow card
    details visible in snippets.
</tool_response>

Step 2: <think>
Search too broad. Need to focus on World Cup
    1990 and 1994 specifically since those had
     Brazilian referees.
</think>
<tool_call name="search">
World Cup 1990 1994 Brazilian referee matches
    yellow cards
</tool_call>
<tool_response>
Found several matches with Brazilian referees.
     Notable: Ireland vs Romania 1990 with
    referee Wright (not Brazilian). Need to
    refine.
</tool_response>

Step 3: <think>
Still not finding Brazilian referee
    specifically. Let me search for Brazilian
    FIFA referees active 1990-1994.
</think>
<tool_call name="search">
Brazilian FIFA referee World Cup 1990 1994
    official matches
</tool_call>
<tool_response>
Found: Jose Wright was actually Jose Roberto
    Wright, Brazilian referee who officiated
    Ireland vs Romania, Italy 1990.
</tool_response>
```

Listing 1: Trajectory showing correct persistence through local error

**Meta-Decision:** CONTINUE - The agent correctly identifies that the search strategy is fundamentally sound but needs refinement. Each step builds logically on the previous, gradually narrowing the search scope.

### E.2 Scenario 1B: Global Dead-End - Correct Revision

This trajectory shows appropriate strategic revision when the current approach hits a fundamental barrier.

```
Step 1-4: [Similar initial searches as above]

Step 5: <think>
Found Jose Roberto Wright as Brazilian referee
    , but need specific yellow card and
    substitution details for Ireland vs
    Romania match.
</think>
<tool_call name="search">
Ireland Romania 1990 World Cup yellow cards
    substitutions details
</tool_call>
<tool_response>
General match reports available, but detailed
    yellow card timing and substitution
    reasons not in search results.
</tool_response>

Step 6-8: [Repeated attempts with similar
    queries yield no detailed card/
    substitution timing]

Step 9: <think>
Search engines not providing the granular
    match details needed. Need to try FIFA
    official records or match reports.
</think>
<tool_call name="search">
site:fifa.com Ireland Romania 1990 World Cup
    match report official
</tool_call>
<tool_response>
FIFA site search limited. No detailed match
    reports available in search results.
</tool_response>
```

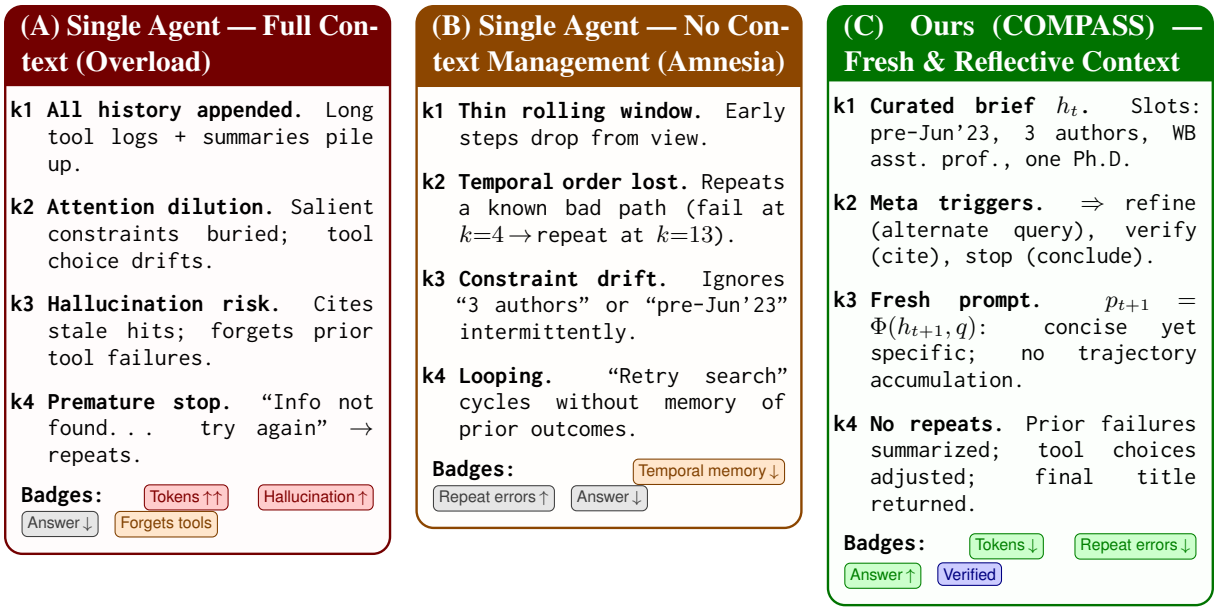


Figure 6: Comparison of agent designs across context management strategies.

```
Step 10: <think>
Current web search approach hitting dead end.
    Need to pivot to historical sports
    databases or newspaper archives from 1990.
</think>
<tool_call name="search">
newspaper archive 1990 Ireland Romania World
    Cup match report yellow cards
</tool_call>
<tool_response>
Found detailed match report from Irish Times
    archive: confirms 4 yellow cards (2 each
    team), 3 in second half, 4 substitutions
    including injury substitution in 23rd
    minute.
</tool_response>
```

Listing 2: Trajectory requiring strategic revision

**Meta-Decision:** REVISE - After steps 6-9 showed repeated failure to find detailed match statistics through general web search, the agent correctly pivoted to historical newspaper archives, which contained the required granular information.

### E.3 Scenario 2A: Correct Solution - Appropriate Conclusion

This trajectory demonstrates proper recognition of task completion.

```
Step 1-7: [Successful search progression
    leading to match identification]

Step 8: <think>
Found comprehensive evidence: Ireland vs
    Romania, 1990 World Cup, Brazilian referee
     Jose Roberto Wright, 4 yellow cards (2
    each team, 3 in 2nd half), 4 substitutions
     including injury sub in 23rd minute. All
    criteria satisfied.
</think>
<tool_call name="verify">
Cross-check: Ireland Romania 1990 World Cup
    referee Jose Roberto Wright Brazilian
</tool_call>
<tool_response>
Confirmed: Jose Roberto Wright, Brazilian
    referee, officiated Ireland 1-0 Romania,
    Italy 1990 World Cup.
</tool_response>

Step 9: <think>
All query requirements verified. Have
    definitive answer with supporting evidence
    .
</think>
<answer>Ireland v Romania</answer>
```

Listing 3: Trajectory with correct conclusion timing

**Meta-Decision:** CONCLUDE - The agent appropriately recognizes that all query constraints have been satisfied and verified, leading to correct termination.

### E.4 Scenario 2B: Incorrect Solution - Necessary Continuation

This trajectory illustrates the importance of continued verification when initial conclusions are flawed.

```
Step 1-5: [Initial searches identify several
    1990-1994 matches with Brazilian referees]

Step 6: <think>
Found Brazil vs Argentina 1990 with Brazilian
    referee. This seems like a clear match -
    Brazilian referee in a Brazil game.
```

```
</think>
<tool_call name="search">
Brazil Argentina 1990 World Cup yellow cards
    substitutions
</tool_call>
<tool_response>
Match report shows 6 yellow cards total, not 4.
     Also only 2 substitutions, not 4 required
    .
</tool_response>

Step 7: <think>
Brazil vs Argentina doesn't match the
    constraints. But wait - would a Brazilian
    referee officiate a Brazil match? That
    seems unlikely for neutrality.
</think>
<tool_call name="search">
FIFA referee neutrality rules Brazilian
    referee Brazil matches World Cup
</tool_call>
<tool_response>
FIFA rules require neutral referees. Brazilian
     referees cannot officiate Brazil national
     team matches.
</tool_response>

Step 8: <think>
My initial assumption was wrong. Need to find
    matches between non-Brazilian teams with
    Brazilian referees.
</think>
<tool_call name="search">
1990 World Cup Brazilian referee non-Brazil
    matches yellow cards
</tool_call>
<tool_response>
Jose Roberto Wright (Brazilian) officiated
    Ireland vs Romania, among others.
</tool_response>
```

Listing 4: Trajectory requiring error recovery through continuation

**Meta-Decision:** CONTINUE - The agent initially reached an incorrect conclusion (Brazil vs Argentina) but through continued analysis discovered the logical flaw in having a Brazilian referee officiate a Brazil match, leading to the correct answer.

### E.5 Failure Pattern Analysis

These trajectories illustrate the two major failure patterns identified in our research:

**Context Overflow:** In scenarios without proper context management, agents accumulate extensive search histories that bury critical constraints (e.g., the "Brazilian referee" requirement becomes lost among dozens of search results about various World Cup matches).

**Premature Termination:** Agents may conclude with superficially plausible but incorrect answers (e.g., Brazil vs Argentina because it involves a Brazilian referee) without proper verification of all constraints.

The COMPASS framework addresses these through explicit meta-monitoring that can trigger strategic revisions (Scenario 1B) or prevent premature conclusions (Scenario 2B) while maintaining efficiency through appropriate continuation decisions (Scenarios 1A and 2A).

### E.6 Agent Complexity Comparison

To illustrate the varying cognitive demands across COMPASS components, we present parallel outputs from all three agents during a single trajectory step. This analysis informed our decision to develop a specialized compact Context Manager.

**Query:** Find the exact publication date of the most recent FDA guidance on machine learning in medical devices.

**Current Step:** Agent has performed 8 search attempts, found multiple FDA documents from different years, but discovered conflicting publication dates across sources.

#### E.6.1 Main Agent Output

The Main Agent performs tactical reasoning with tool selection:

```
<think>
Found FDA documents from 2021, 2022, and 2023,
     but dates are inconsistent across sources
    .
The 2023 document mentions "updated guidance"
    but some sources show 2022 as most recent.
Need to verify directly from FDA website to
    get authoritative publication date.
Should search specifically on FDA.gov domain
    to avoid third-party date errors.
</think>
<tool_call name="search">
site:fda.gov "machine learning" "medical
    devices" guidance 2023 publication date
</tool_call>
```

Listing 5: Main Agent tactical execution

**Analysis:** The Main Agent focuses on immediate next steps, tool selection, and tactical problem-solving. Its reasoning is linear and execution-focused.

#### E.6.2 Meta-Thinker Output

The Meta-Thinker performs strategic oversight and anomaly detection:

```
<analysis>
TRAJECTORY ASSESSMENT:
- 8 search attempts completed
- Multiple candidate documents identified
    (2021, 2022, 2023)
```

```
- ANOMALY DETECTED: Date inconsistencies
    across sources suggest information quality
     issues
- RISK: Agent may conclude with incorrect date
     due to conflicting evidence

STRATEGIC EVALUATION:
Current approach: Appropriate focus on
    authoritative FDA source
Persistence vs Pivot: CONTINUE - strategy is
    sound, execution refining toward official
    source
Completion readiness: NOT READY - conflicting
    evidence requires verification

INTERVENTION DECISION: VERIFY
Rationale: Before concluding, agent should
    cross-reference publication metadata
from official FDA document rather than relying
     on search result summaries.

MONITORING PRIORITY: High - date accuracy
    critical for regulatory information
</analysis>
```

Listing 6: Meta-Thinker strategic analysis

**Analysis:** The Meta-Thinker engages in complex strategic reasoning, anomaly detection, risk assessment, and multi-level decision making about trajectory management.

### E.6.3 Context Manager Output

The Context Manager performs structured summarization and context synthesis:

```
UPDATED CONTEXT BRIEF:

OBJECTIVE: Find exact publication date of most
     recent FDA ML medical device guidance

SEARCH PROGRESS:
- Completed: 8 search iterations
- Sources explored: FDA documents, medical
    journals, regulatory news sites
- Candidate documents identified:
  * 2021: "Software as Medical Device"
    guidance
  * 2022: "Machine Learning-Enabled Medical
    Devices" draft
  * 2023: "Updated ML/AI Medical Device
    Guidance" (date disputed)

CURRENT FINDINGS:
 Confirmed: FDA has published ML medical
    device guidance
 Unresolved: Exact publication date of most
    recent version (2022 vs 2023 conflict)

KEY CONSTRAINTS:
- Must be official FDA publication (not draft
    or third-party summary)
- Need authoritative publication date, not
    revision or comment period dates
- Focus on "medical devices" specifically (not
     general AI/ML guidance)

NEXT STEP CONTEXT:
Agent attempting FDA.gov domain search to
    resolve date conflict.
Priority: Verify official publication metadata
     from primary source.
```

Listing 7: Context Manager context synthesis

**Analysis:** The Context Manager performs systematic information organization, constraint tracking, and structured summarization. Its output follows predictable templates and focuses on factual synthesis rather than strategic reasoning.