DataSeer: A Manager-Centric Collaborative Multi-Agent Framework with Multi-Branch Reasoning for Automated Insight Discovery

Suchen Liu[1,2], **Yuanfeng Song**[2,*], **Jun Gao**[1,*], **Xing Chen**[2]
[1]Key Laboratory of High Confidence Software Technologies, CS, Peking University, China
[2]ByteDance, China

Abstract

The increasing complexity of data drives the demand for automated insight discovery. While LLMs and agent technologies have advanced data analysis, existing methods struggle with maintaining contextual coherence, achieving sufficient coverage (due to single-path exploration), and adapting rigid planning to dynamic data discovery. We propose DataSeer, a collaborative multi-agent framework for automated insight discovery. Our first contribution is a **Manager-Centric Collaborative Framework**, where the Manager ensures cross-episode contextual coherence through a dual-layer memory system with compression, consolidation, and retrieval, alongside dynamic prompt editing, coordinating the overall process between the Planner and Executor. Second, we **optimize the planning and execution components**: the Planner employs multi-role discussion for adaptive sub-goal generation and plan refinement; the Executor is endowed with tactical autonomy for exploratory execution and incorporates real-time multi-dimensional self-assessment to guarantee insight quality. Third, we design **Multi-Branch Reasoning** that executes multiple discovery trajectories and synthesizes outcomes through LLM-based aggregation, improving coverage and reducing single-path stochasticity. Experiments on InsightBench and InsightEval show that DataSeer outperforms baselines, achieving improvements of 18.7% and 12.1% in insight-level scores, and 11.6% and 10.3% in summary-level scores, respectively.

1 Introduction

In the current data-driven world, organizations are inundated with complex datasets, yet often lack the resources to uncover profound insights. Fortunately, advances in Large Language Models (LLMs) and intelligent agent technologies are closing this gap by enabling automated insight discovery (Yao et al., 2023; OpenAI, 2024; Guo et al., 2025). LLM-based agents can autonomously query data, identify patterns, and generate reports, transforming raw data into actionable intelligence (Sahu et al., 2025; Lu et al., 2025; Lei et al., 2025; Cheng et al., 2023; Zhu et al., 2026; Lu et al., 2026).

However, leveraging LLMs for automated insight generation inherently involves open-ended objectives, complex multi-step dependencies, and long-chain reasoning. When addressing such insight discovery tasks, existing methods often exhibit three fundamental limitations. First, **insufficient contextual coherence**: they are prone to context fragmentation and memory decay across multiple exploration rounds, resulting in a lack of continuity throughout the analytical process (Maharana et al., 2024). Second, **weak adaptability of planning mechanisms**: fixed planning templates struggle to adapt to the dynamic nature of data exploration, often leading to a disconnect between planning and execution (Guan et al., 2025). Third, **limited coverage of the reasoning path**: reliance on single-path sequential decision-making can easily lead to local optima, causing inadequate insight coverage (Wei et al., 2025). These limitations constrain the quality of the generated insights.

To systematically address the above-mentioned challenges, we propose DataSeer, a collaborative multi-agent framework for automated insight discovery. Our core contribution is **a hierarchical collaborative framework** comprising three specifically designed agents: **the Manager**, **the Planner**, and **the Executor**. Among them, to address Limitation 1, the Manager mitigates context fragmentation both through a dual-layer memory system (with compression, consolidation, and retrieval) and through dynamic prompt editing, ensuring cross-episode coherence; To mitigate Limitation 2, the Planner employs multi-role discussion and reflection mechanisms, generating comprehensive sub-goals and execution plans; The Executor flexibly executes plans with tactical auton-

*Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2026, pages 16030–16058

omy, while conducting real-time multi-dimensional self-evaluation during exploration to ensure output insight quality. Furthermore, to alleviate Limitation 3, we introduce **a Multi-Branch Reasoning** mechanism that mitigates single-path bias by executing and merging multiple analytical trajectories, thereby enhancing the insight quality.

In a nutshell, our contributions can be summarized as follows:

- We propose DataSeer, a **Manager-Centric Collaborative Framework**. The Manager ensures contextual coherence through a dual-layer memory system and dynamic prompt editing, coordinating the overall discovery process between the Planner and Executor.
- We optimize **the planning and execution components**: the Planner employs multi-role discussion for adaptive sub-goal generation and plan refinement; the Executor is endowed with tactical autonomy for exploratory execution and incorporates real-time multi-dimensional self-assessment to guarantee the quality and relevance of generated insights.
- We design **Multi-Branch Reasoning** that executes multiple discovery trajectories and synthesizes outcomes through LLM-based aggregation, improving coverage and reducing single-path stochasticity.

Extensive experiments on InsightBench and InsightEval show that DataSeer significantly outperforms strong baselines, validating the effectiveness of our approach.

2 Related Work

In this section, we review the literature foundation of our work. We first discuss the evolution of LLM-based agent architectures, from single-agent reasoning to multi-agent collaboration. Next, we examine agent-driven insight discovery approaches and their limitations in open-ended exploration. Finally, we review data agent benchmarks, focusing on evaluation frameworks for exploratory analysis and insight generation.

Agent Architectures. LLM-based agents initially center around single-agent interaction loops, in which a single model alternates between reasoning steps and environment interactions. Within this setting, ReAct (Yao et al., 2023) exemplifies how integrating intermediate reasoning with tool invocation facilitates more reliable and actionable step-wise reasoning. Follow-up studies enrich this paradigm by injecting additional control mechanisms, such as self-evaluative feedback (Shinn et al., 2023) or explicit plan construction (Wang et al., 2023), to mitigate instability as task horizons grow.

As task complexity increases, research has shifted toward multi-agent architectures that distribute reasoning and exploration across cooperating agents. Role-driven communication schemes (Li et al., 2023) and workflow-oriented coordination frameworks (Hong et al., 2023) illustrate how specialization and structured interaction can improve scalability beyond single-agent limits. Moving further away from fixed-goal execution, systems such as Voyager (Wang et al., 2024) highlight continual exploration with persistent skill acquisition in interactive environments. More recent efforts, including Aime (Shi et al., 2025) and WebWeaver (Li et al., 2025), investigate adaptive coordination patterns that interleave planning, execution, and synthesis to support open-ended reasoning processes. However, maintaining contextual coherence and coordinated planning across multiple exploration rounds remains challenging, especially when analytical goals evolve dynamically.

Insight Discovery. Agent-based insight discovery targets open-ended exploration without predefined analytical goals, emphasizing the iterative surfacing and organization of informative patterns. At the tooling level, Pandas Agent (LangChain, 2024) enables LLM-driven exploratory analysis via executable Pandas operations. At the system level, InsightPilot (Ma et al., 2023) integrates LLM reasoning with analytical engines, where components such as QuickInsights, MetaInsight, and XInsight support different strategies for generating and structuring candidate insights. Related designs like AgentPoirot (Sahu et al., 2025) further formalize insight discovery as an iterative loop of question generation, analysis execution, and insight synthesis. Despite these advances, existing approaches largely rely on fixed heuristics or shallow exploration strategies, which may lead to limited coverage and contextual fragmentation across extended analytical sessions, ultimately constraining the depth and breadth of generated insights.

Data Agent Benchmarks. Evaluation of data analysis agents has largely relied on task-oriented benchmarks with explicit objectives and deterministic correctness criteria. Representative examples include DS-1000 (Lai et al., 2023) for data science

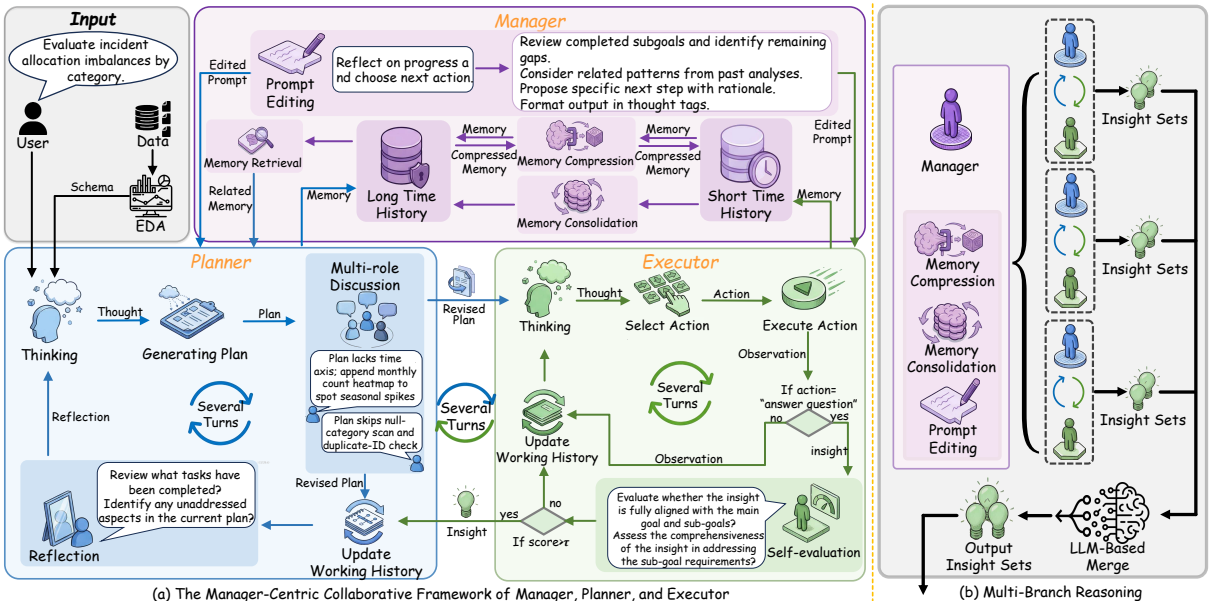

Figure 1: Architecture of DataSeer, a manager-centric collaborative multi-agent framework for automated insight discovery. The system integrates two core components: **(a) The Manager-Centric Collaborative Framework of Manager, Planner, and Executor** and **(b) Multi-Branch Reasoning**. (a) The Manager ensures contextual coherence through a dual-layer memory system and dynamic prompt editing; the Planner generates adaptive sub-goals and execution plans via multi-role discussion; and the Executor conducts tactical exploration with autonomous decision-making and multi-dimensional self-assessment. (b) The MBR component runs multiple discovery trajectories and aggregates their outcomes through LLM-based synthesis to mitigate single-path stochasticity and enhance insight quality.

code generation, Text2analysis for end-to-end "raw-text-to-insight" pipelines, and multi-step benchmarks such as DABench (Hu et al., 2024). Since our focus is on open-ended insight discovery, where analytical goals may evolve during exploration, we adopt benchmarks specifically designed for this setting: InsightBench (Sahu et al., 2025) and InsightEval (Zhu et al., 2026). Furthermore, these benchmarks expose the challenges of maintaining coherent exploration trajectories and assessing multi-faceted insight quality in truly open-ended scenarios, highlighting the need for frameworks that can sustain goal-directed reasoning while adapting to emerging analytical paths.

3 Methodology

The DataSeer framework introduces automated insight discovery via three complementary mechanisms: (1) a **Manager-Centric Collaborative Framework** that enables hierarchical task coordination through dynamic coordination; (2) **Optimized Planning and Execution** through multi-role discussion and tactical autonomy with integrated quality assessment; and (3) **Multi-Branch Reasoning** that ensures robust discovery through parallel exploration and synthesis.

3.1 The Manager-Centric Collaborative Framework: Manager, Planner, and Executor

The manager-centric collaborative framework of Manager, Planner, and Executor (MPE) represents the core analytical engine of DataSeer, tackling key challenges in automated insight discovery: the Manager mitigates context fragmentation through memory management and prompt optimization; the Planner overcomes rigid planning via adaptive strategy generation; and the Executor ensures reliable insights through tactical exploration with integrated quality assessment.

3.1.1 Manager: Centralized Memory Management and Contextual Coordination

The Manager serves as the system's central coordinator, responsible for maintaining cross-episode continuity through integrated memory management and dynamic prompt optimization. By implementing a sophisticated memory system with compression, retrieval, and consolidation mechanisms, alongside context-aware prompt editing, the Manager ensures that the Planner and the Executor operate with both historical awareness and task-specific adaptation.

Memory Storage and Compression. The Manager maintains two primary memory stores with distinct roles and data sources:

- **Short-Term Memory (STM, $\mathcal{M}^{\text{STM}}$)**: A transient buffer that captures the immediate outcomes of successful investigative actions. Specifically, STM stores the observation O_i by the **Executor** during its tactical exploration of sub-goals.

- **Long-Term Memory (LTM, $\mathcal{M}^{\text{LTM}}$)**: A consolidated, generalized repository of compressed memory units. LTM directly stores high-level, abstracted knowledge, including the **Planner**'s analytical thoughts T_i, execution plans P_i, and reflective assessments R_i. Additionally, LTM receives and integrates distilled content from STM through the consolidation process.

Both STM and LTM employ a *compression mechanism* when they store memory. This process transforms analytical outputs into structured memory units through abstraction, reducing storage overhead while preserving semantic content.

For the Executor's findings stored in STM, each observation O_i undergoes compression to create a memory unit m_i^{exec}. The compression operates at two levels: preserving core observations (Level 1) and generalizing into transferable patterns (Level 2). Similarly, for the Planner's strategic content stored in LTM, the analytical thoughts T_i, execution plans P_i, and reflections R_i are compressed into memory units m_j^{plan}, focusing on extracting strategic essence and methodological insights.

Memory Consolidation and Retrieval. After each plan-execute episode, the system performs *memory consolidation* to integrate the Executor's tactical findings from STM into the Planner's strategic memory in LTM. During consolidation, each compressed memory unit m_i^{exec} from $\mathcal{M}^{\text{STM}}$ is compared with existing strategic units m_j^{plan} in $\mathcal{M}^{\text{LTM}}$ through semantic similarity calculation.

When $\text{Sim}(m_i^{\text{exec}}, m_j^{\text{plan}}) > \theta$ ($\theta = 0.7$), the Executor's concrete finding aligns with and refines a pre-existing strategic pattern from the Planner. The system then merges m_i^{exec} and m_j^{plan} to form a new, enriched unit $m_{j'}^{\text{plan}}$ that encapsulates both strategic intent and empirical evidence. If no sufficiently similar strategic unit exists, m_i^{exec} is added as a new, standalone entry to LTM, potentially seeding a new line of strategic thought.

During each thinking step of the Planner, the Manager dynamically retrieves contextual support from the consolidated knowledge base in LTM. This retrieval is triggered by the Planner's current strategic context, encoded as a query embedding q that incorporates the analytical landscape.

The system retrieves the top-k relevant memory units from LTM based on semantic similarity:

$$\mathcal{H}_{\text{mem}} = \underset{m_j \in \mathcal{M}^{\text{LTM}}}{\text{Top-}k} \ \text{Sim}(m_j, q) \tag{1}$$

where $\mathcal{H}_{\text{mem}}$ is the retrieved memory context provided to the Planner.

Dynamic Prompt Editing for Task-Specific Adaptation. The use of generic action templates across diverse analytical tasks presents a fundamental limitation: standardized templates may not fully align with the specific contextual requirements of individual investigative scenarios. To address this challenge, the Manager implements targeted prompt editing during execution phases of both the Planner and Executor, adapting action templates to better match current task demands.

This approach operates through two complementary mechanisms:

- **Format-Preserving Rewriting**: The Manager performs constrained prompt modifications that preserve original input-output formats of all action templates, ensuring workflow stability. This allows components to continue operating within established procedural frameworks while benefiting from contextually-optimized instructions.

- **Context-Driven Adaptation**: Each prompt rewrite incorporates three dimensions of contextual information:
  - **Current Step Objective**: The immediate sub-goal G_i for execution phases or global goal G for planning phases
  - **Task Type Characteristics**: Analytical patterns and requirements specific to the investigation domain
  - **Contextual History**: The evolving discovery trajectory captured in $\mathcal{H}_{\text{Planner}}$ and $\mathcal{H}_{\text{mem}}$ for the Planner and $\mathcal{H}_{\text{Executor}}$ for the Executor

This contextual integration allows the system to align prompts with both immediate analytical needs and overarching discovery goals. During planning, prompt editing ensures strategic reasoning incorporates relevant methodological guidance; during execution, prompt editing calibrates tactical operations to the specific investigative landscape. Thus, by ensuring strategic alignment during planning and enabling tactical adaptation during execution, prompt editing directly mitigates context fragmentation and maintains a coherent analytical thread.

3.1.2 Planner: Progressive Guidance through Multi-Role Discussion and Reflection

The Planner addresses fundamental limitations in analytical planning methodologies by synthesizing strategic foresight with adaptive refinement. Traditional methods often suffer from either rigidity in predetermined task decomposition or insufficient direction in purely reactive paradigms. Our Planner navigates this trade-off through a sophisticated mechanism that dynamically identifies optimal investigation pathways based on cumulative discovery history, then generates precise guidance and execution plans for each identified sub-goal.

At the core of this approach lies a structured reasoning process grounded in the *ReAct (Reasoning-Acting)* framework. The Planner initiates each cycle with analytical deliberation, systematically evaluating the current analytical landscape, before transitioning to concrete plan formulation.

The Planner's contextual foundation is synergistically assisted by the Manager. The Planner naturally maintains a linear working history $\mathcal{H}_{\text{Planner}} = [\ldots, (T_{i-1}, \hat{P}_{i-1}, O_{i-1}), \ldots]$, which provides coherent continuity by recording the sequence of its own thoughts (T), revised plans ($\hat{P}$), and validated observation insights (O) from the Executor. This self-record is cross-referenced with the memory $\mathcal{H}_{\text{mem}}$ retrieved by the Manager, which stores generalized patterns and distilled experiences from analogous historical steps. During each reasoning step, the Planner cooperatively queries both contexts: $\mathcal{H}_{\text{Planner}}$ ensures a logically consistent next step within the current task narrative, while $\mathcal{H}_{\text{mem}}$ injects relevant tactical knowledge, provides a global task perspective by accurately referencing historical execution outcomes from the Executor, and safeguards against past pitfalls.

Adaptive Strategy Planning Cycle. Formally, the planning process operates as a multi-phase mechanism. Let $i = 1, 2, \ldots$ index the planning–execution round. The Planner receives the global goal G and the data schema S at the first round. The Planner acquires its evolving memory, denoted uniformly as $\mathcal{H}_{\text{mem}}$, via Memory Retrieval from the Manager at every step. The cycle comprises the following stages:

During its initial cycle, The Planner engages in a reasoning process from which the reflective component R is available after the first round.

$$T_i = \begin{cases} \Pi_{\text{Think}}\big(G, S, \mathcal{H}_{\text{mem}}\big), & i = 1 \\ \Pi_{\text{Think}}\big(G, S, \mathcal{H}_{\text{mem}}, R_{i-1}\big), & i \geq 2 \end{cases} \tag{2}$$

This thinking phase produces analytical thoughts T_i that capture the strategic rationale for subsequent actions. These thoughts then inform the generation of a targeted guidance package:

$$(G_i, P_i) = \Pi_{\text{Plan}}(T_i, G, S, \mathcal{H}_{\text{mem}}, \mathcal{H}_{\text{Planner}}) \tag{3}$$

where the sub-goal G_i has a corresponding execution plan P_i detailing methods to investigate G_i.

To ensure robustness and feasibility, the initial plan undergoes rigorous refinement through *multi-role discussion*. This collaborative process engages diverse analytical perspectives to evaluate plan coherence, identify potential oversights, and validate alignment with schema constraints. The discussion synthesizes these perspectives to produce an optimized execution strategy:

$$\hat{P}_i = \Pi_{\text{Discussion}}(P_i, T_i, G, S, \mathcal{H}_{\text{Planner}}) \tag{4}$$

Following plan execution by the Executor, which yields validated observation insights O_i of the sub-goal G_i, the system engages in a comprehensive *reflection* phase. This critical evaluation examines the entire analytical trajectory, encompassing reasoning processes, plan formulation, execution methodologies, and outcome quality, to extract insights and identify improvement opportunities:

$$R_i = \Pi_{\text{Reflection}}(G, S, O_i, \mathcal{H}_{\text{Planner}}, \mathcal{H}_{\text{mem}}) \tag{5}$$

The working history is updated each cycle to incorporate the latest reasoning step and outcome:

$$\mathcal{H}_{\text{Planner}} \leftarrow \mathcal{H}_{\text{Planner}} \cup (T_i, \hat{P}_i, O_i) \tag{6}$$

This structured cycle, which includes thinking, plan generation, multi-role discussion, plan execution, and reflective assessment, creates a self-improving analytical engine. Each iteration builds on the accumulated insights while maintaining strategic alignment with the overarching objectives.

3.1.3 Executor: Autonomous Tactical Investigation with Integrated Quality Assessment

The Executor is an agile tactical agent dedicated to focused investigation within each sub-goal. At the start of each sub-goal, the Executor's working history $\mathcal{H}_{\text{Executor}}$ is initialized as an empty sequence to maintain task focus. Unlike the Planner, the Executor operates without retrieving Manager memory, concentrating exclusively on tactical execution.

To maintain alignment with overarching objectives, the Executor receives both the immediate sub-goal G_i and the global goal G, preventing exploratory drift. Upon receiving the Planner's guidance (G_i and execution plan $\hat{P}_i$), the Executor actively interprets and adapts the plan based on real-time findings, autonomously formulating tactical sub-tasks to fulfill the sub-goal. The Executor operates through two interconnected phases: a continuous tactical exploration loop and a conditional quality assessment mechanism.

Autonomous Tactical Exploration Cycle. The Executor runs a continuous exploration loop for autonomous tactical investigation within each sub-goal, maintaining persistent context through five sequential steps:

1. **Generate Thought**: The Executor begins each iteration with analytical deliberation, generating tactical insights about how to approach the current exploration state:

$$T_{i,j} = \Pi_{\text{Think}}(G_i, \hat{P}_i, G, S, \mathcal{H}_{\text{Executor}}) \quad (7)$$

   This thinking produces strategic thoughts $T_{i,j}$ that inform subsequent action selection, ensuring alignment with both the Planner's strategic direction and the evolving discovery context.

2. **Select Action**: Building upon the thinking phase, the Executor selects an action a_j from a comprehensive action space:

   - *GenerateQuestion*: Pose new questions from goals or previous questions
   - *SelectQuestion*: Choose the most promising question
   - *AnswerQuestion*: Answer selected question using dataset capabilities
   - *Summarize*: Synthesize key findings from accumulated insights
   - *Halt*: Decide analysis completion and terminate exploration

   This action space embodies three complementary capabilities: *path planning* (GenerateQuestion, SelectQuestion) that structures the investigative approach, *problem-solving* (AnswerQuestion) that generates concrete analytical results, and *control flow* (Summarize, Halt) that manages exploration termination.

3. **Execute Action**: The Executor executes the selected action and observes the outcomes. For path planning actions, execution involves updating the question hierarchy and exploration strategy; for problem-solving actions, execution entails computational operations and data analysis that generate observation $\hat{o}_{i,j}$.

4. **Process Observation**: The Executor processes the observation results:

   - For path planning and control actions, the Executor directly proceeds to update the state.
   - For *AnswerQuestion* actions, observations trigger the quality assessment mechanism.

5. **Update History**: At the end of each cycle iteration, the Executor updates its working history by appending the current thinking trace:

$$\mathcal{H}_{\text{Executor}} \leftarrow \mathcal{H}_{\text{Executor}} \cup (T_{i,j}, a_j, \hat{o}_{i,j}) \quad (8)$$

   where $o_{i,j}$ denotes the observation from executing action a_j. This historical record enables the Executor to avoid repetitive exploration and maintain continuity within the sub-goal.

Conditional Quality Assessment Mechanism. When the action is *AnswerQuestion*, the Executor activates a separate quality assessment mechanism to ensure only high-quality insights are validated:

1. **Self-evaluation**: Executor scores the answer on confidence, novelty, and diversity.

$$s_{i,j} = \Pi_{\text{Score}}(\hat{o}_{i,j}, \mathcal{H}_{\text{Executor}}) \quad (9)$$

2. **Check Threshold**: Executor checks the quality score against preset threshold.

   - If $s_{i,j} > \theta$ (where θ is a predefined quality threshold), the insight $O_i = \hat{o}_{i,j}$ is returned to the Planner as a validated discovery, and the current execution cycle concludes.
   - If $s_{i,j} \leq \theta$, the Executor continues the exploration loop, using the feedback to refine subsequent approaches.

The Executor proceeds through multiple iterations of this tactical exploration loop, maintaining persistent context and autonomously adapting its strategy based on cumulative findings. This autonomous tactical investigation continues until either producing a validated insight through the quality assessment mechanism or autonomously deciding to halt via the control actions.

This design grants the Executor autonomy to explore, interpret, and adapt within its designated scope. Crucially, this exploratory freedom is counterbalanced by a built-in validation step: insights generated via *AnswerQuestion* must pass a multi-dimensional quality assessment before being deemed valid for the Planner. This ensures that the autonomy afforded to the Executor consistently translates into high-quality, relevant findings, making it an effective tactical investigator that robustly complements the Planner's strategic guidance.

3.1.4 Integrated Collaboration Dynamics

The three components interact through a strictly coordinated and deterministic dynamic process: the Planner and Executor engage in a coupled iterative loop of planning and execution, while the Manager orchestrates memory operations and prompt optimization through a rule-based scheduling policy. To ensure system reproducibility and stable control flow, the Manager's core operations are triggered synchronously at fixed points within the loop, rather than relying on stochastic or non-deterministic orchestration.

Specifically, the collaboration follows these deterministic schedules, as depicted in Figure 1:

- **Memory Retrieval** occurs synchronously at fixed stages (i.e., during thought generation, plan generation, and reflection).
- **STM Updates** execute synchronously immediately after each Executor action.
- **Memory Consolidation and LTM Updates** are triggered synchronously at the conclusion of each complete Plan-Execute round.

While the semantic compression of stored memory entries is performed asynchronously in the background to optimize system latency, this process is semantically lossless. Whether a compression task finishes before or after a subsequent retrieval event, the semantic content remains consistent. Consequently, the asynchronous nature of compression does not introduce randomness; the control flow remains entirely deterministic and dictated by the fixed Plan-Execute loop.

Together, this design creates a virtuous cycle in which strategic planning directs tactical execution, execution outcomes feed back to refine strategic understanding, and centralized, deterministic coordination by the Manager ensures persistent contextual coherence across the entire discovery process. The specialized innovations of each component complement one another to form a robust, integrated system that systematically addresses the core challenges of automated insight discovery.

3.2 Multi-Branch Reasoning for Robust Insight Discovery

Even with adaptive planning and quality-aware execution, a single discovery trajectory may still miss meaningful insights due to stochastic biases or narrow focus. The **Multi-Branch Reasoning** mechanism addresses this limitation by conducting multiple parallel Plan–Execute trajectories, then aggregating outcomes to enhance insight quality.

The MBR module executes K independent reasoning branches, where each branch k operates with distinct configurations, while incorporating insights from prior branches:

$$\{(G_i^{(k)}, \hat{P}_i^{(k)}, O_i^{(k)})\}_{k=1}^{K} \tag{10}$$

This execution with knowledge transfer by Meta-Agent enables continuous refinement, as later branches explicitly build upon and verify earlier findings. The diversified initialization strategies ensure each branch explores different regions of the analytical space, while the incorporation of prior results minimizes redundant computation and allows subsequent branches to focus on promising or under-explored directions.

After all branches complete their discovery processes, their outcomes are merged through a large language model-based synthesis:

$$O^* = \Pi_{\text{LLM-Merge}}(\{O_i^{(k)}\}_{k=1}^{K}) \tag{11}$$

The LLM-based merge operator LLM-Merge performs several critical functions: it identifies and reinforces recurring themes across branches, integrates complementary insights, resolves contradictions through evidence-weighted reasoning, and filters out low-quality outputs through semantic understanding rather than relying solely on numerical thresholds. This design ensures that reliable insights are reinforced through multi-branch

agreement, while novel and diverse findings from individual branches are preserved. MBR thus effectively mitigates single-path stochasticity, enhances evidence grounding through cross-validation, and improves the coverage of the discovered insights.

4 Experiments

In this section, we conduct comprehensive experiments to evaluate the effectiveness of DataSeer. We first detail the experimental setup, including the datasets, baselines, and evaluation metrics. Next, we present the main results to demonstrate the superiority of DataSeer in insight discovery compared to strong baselines. Finally, we perform thorough ablation studies to validate the individual contributions of our core components.

4.1 Experimental Setup

We briefly summarize the experimental setup, with more details of evaluation metrics, implementation details and experimental results provided in Appendix A.

Dataset. We evaluated our approach using two widely adopted datasets: InsightBench (Sahu et al., 2025) and InsightEval (Zhu et al., 2026).

- **InsighBench** (Sahu et al., 2025): InsighBench is a widely used benchmark for evaluating insight discovery in data analytics. It consists of 100 tabular datasets representing diverse business use cases, spanning three difficulty levels: easy, medium, and hard. Unlike other datasets that focus on more specific QA style data analysis tasks (Hu et al., 2024; Majumder et al., 2025), InsightBench evaluates agents on their ability to perform end-to-end data analytics, encompassing question formulation, answer interpretation, and insight generation.

- **InsighEval** (Zhu et al., 2026): InsightEval is an expert-curated benchmark based on InsighBench to rigorously assess the insight-discovery capabilities of LLM-driven data agents. It comprises 100 high-quality analysis instances with 1000 insights spanning eight business domains (e.g., Incident Management, Asset Management). Each instance pairs a CSV table with a precisely defined analytical goal and is associated with ten expert-validated questions covering six insight types (Descriptive, Diagnostic, Predictive, Prescriptive, Evaluative, and Exploratory), along with concise reference insights and a synthesized summary. The benchmark is structured across four progressive difficulty levels to enable comprehensive evaluation.

Baselines. We compare DataSeer with many strong baselines from the literature to demonstrate its superior insight-discovery performance. The details of these baselines are the following:

- **ReAct** (Yao et al., 2022): ReAct is the Reasoning-Acting framework that interleaves reasoning steps with actions, adapted for data analysis tasks.

- **Pandas Agent** (LangChain, 2024): A LangChain-based conversational agent for natural language interaction with pandas DataFrames, translating queries into sequential pandas operations.

- **AgentPoirot** (Sahu et al., 2025): AgentPoirot is an agent system specifically designed for insight discovery, featuring specialized agent roles for hypothesis generation, data exploration, and insight validation.

- **Aime** (Shi et al., 2025): Aime is a multi-agent framework that eschews rigid "plan-then-execute" in favor of continuous replanning, on-demand agent creation, and shared state tracking through a central module.

Evaluation Metrics. Following the evaluation metircs in InsightEval (Zhu et al., 2026) and InsightBench (Sahu et al., 2025), we employ G-Eval by gpt-4o for automated assessment, reporting both **Insight-level Scores** (individual insight quality) and **Summary-level Scores** (overall analytical narrative quality) across three difficulty levels.

Implementation Details. By default, the base model of DataSeer and baselines is **DeepSeek-V3-250324** (temperature=0). In DataSeer, Planner and Executor run 10 cycles, and Multi-Branch Reasoning uses 2 branches. Evaluation employs **gpt-4o-2024-11-20** for G-Eval scoring.

4.2 Main Results

Comprehensive evaluation on InsightBench and InsightEval (Tables 1 and 2) demonstrates the superior performance of DataSeer. DataSeer reaches summary-level scores of 0.6048 and 0.6901 (exceeding the best baseline by 11.6% and 10.3%) and insight-level scores of 0.5117 and 0.5460 (leading

Model	Insight-level Scores (G-Eval)				Summary-level Scores (G-Eval)			
	Easy	Medium	Hard	Avg	Easy	Medium	Hard	Avg
ReAct (Yao et al., 2022)	0.4355	0.36487	0.3140	0.3698	0.5644	0.5057	0.4936	0.5194
Pandas Agent (LangChain, 2024)	0.4826	0.4036	0.2911	0.3913	0.5534	0.5343	0.4278	0.5059
AgentPoirot (Sahu et al., 2025)	0.5113	0.4295	0.3579	0.4311	0.4740	0.4476	0.3584	0.4270
Aime (Shi et al., 2025)	0.4331	0.3763	0.3149	0.3737	0.5942	0.5269	0.5100	0.5417
DataSeer (Ours)	**0.5852**	**0.4877**	**0.4712**	**0.5117**	**0.6240**	**0.6102**	**0.5803**	**0.6048**

Table 1: Performance Analysis of Various Methods on InsightBench

Model	Insight-level Scores (G-Eval)				Summary-level Scores (G-Eval)			
	Easy	Medium	Hard	Avg	Easy	Medium	Hard	Avg
ReAct (Yao et al., 2022)	0.4666	0.4492	0.3887	0.4351	0.5742	0.5574	0.5501	0.5600
Pandas Agent (LangChain, 2024)	0.4552	0.4441	0.3948	0.4317	0.5324	0.5340	0.5420	0.5361
AgentPoirot (Sahu et al., 2025)	0.5382	0.4948	0.4509	0.4937	0.5762	0.5560	0.5234	0.5517
Aime (Shi et al., 2025)	0.5105	0.4653	0.4029	0.4589	0.6527	0.6142	0.6140	0.6257
DataSeer (Ours)	**0.5971**	**0.5455**	**0.5227**	**0.5537**	**0.7187**	**0.6860**	**0.6683**	**0.6901**

Table 2: Performance Analysis of Various Methods on InsightEval

Model	Insight-level Scores (G-Eval)				Summary-level Scores (G-Eval)			
	Easy	Medium	Hard	Avg	Easy	Medium	Hard	Avg
DataSeer	**0.5852**	**0.4877**	**0.4712**	**0.5117**	**0.6240**	**0.6102**	**0.5803**	**0.6048**
DataSeer w/o MPE	0.5522	0.4700	0.4077	0.4747	0.5289	0.5102	0.4641	0.5011
DataSeer w/o MBR	0.5269	0.4686	0.4379	0.4762	0.5967	0.5771	0.5570	0.5765
DataSeer w/o MPE, MBR	0.5113	0.4295	0.3579	0.4311	0.4740	0.4476	0.3584	0.4270

Table 3: Ablation Study of DataSeer on InsightBench

Model	Insight-level Scores (G-Eval)				Summary-level Scores (G-Eval)			
	Easy	Medium	Hard	Avg	Easy	Medium	Hard	Avg
DataSeer	**0.5971**	**0.5455**	**0.5227**	**0.5537**	**0.7187**	**0.6860**	**0.6683**	**0.6901**
DataSeer w/o MPE	0.5643	0.5170	0.4862	0.5213	0.7029	0.6684	0.6233	0.6643
DataSeer w/o MBR	0.5666	0.5249	0.4947	0.5278	0.6461	0.6491	0.6685	0.6544
DataSeer w/o MPE, MBR	0.5382	0.4948	0.4509	0.4937	0.5762	0.5560	0.5234	0.5517

Table 4: Ablation Study of DataSeer on InsightEval

by 18.7% and 12.1%), with consistent advantages across all difficulty levels.

4.3 Ablation Study

Ablation studies on InsightBench (Table 3) and InsightEval (Table 4) quantify the contribution of each core component. Removing the MPE framework reduces insight-level scores by 7.2% and 5.9%, and summary-level scores by 17.1% and 3.7%, respectively. Disabling MBR leads to decreases of 6.9% and 4.7% in insight-level scores, and 4.7% and 5.2% in summary-level scores. The variant without both components (i.e., AgentPoirot) exhibits significantly lower performance across all metrics, confirming that the integrated design of MPE and MBR is essential for DataSeer's improved insight quality. Notably, MPE has a stronger impact on summary-level coherence, underscoring its role in maintaining narrative continuity, while MBR contributes more evenly to both insight-level and summary-level coverage by mitigating single-path stochasticity.

5 Conclusion

This paper presents DataSeer, a manager-centric multi-agent framework for automated insight discovery. We make three key contributions: (1) a Manager-Centric Collaborative Framework where the Manager ensures contextual coherence through a dual-layer memory system and dynamic prompt editing, coordinating the Planner and Executor; (2) optimized planning and execution, with the Planner employing multi-role discussion for adaptive subgoal generation and the Executor endowed with tactical autonomy and real-time multi-dimensional self-assessment; and (3) Multi-Branch Reasoning that mitigates single-path bias through parallel exploration and LLM-based synthesis. Experiments on InsightBench and InsightEval validate the effectiveness of our approach.

Limitations

- **Dependency on LLM Capabilities:** The performance of DataSeer is inherently tied to the underlying LLM's reasoning and code-generation abilities. While we observe consistent gains across two different LLMs (GPT-4o and DeepSeek), the framework may still inherit known LLM limitations such as occasional hallucination, sensitivity to prompt phrasing, and difficulty with very long-context tasks.

- **Computational Overhead and Response Latency.** To enhance reasoning accuracy, our novel architecture incorporates multiple rounds Reasoning Paths. However, like most current agent frameworks, these mechanisms inevitably introduce significant token overhead and computational costs. A promising research avenue to mitigate this issue lies in adaptive computation or lightweight reasoning distillation, which opens up a compelling new dimension for efficiency optimization. We intend to explore this direction as the next phase of our research to achieve a better balance between systemic intelligence and operational efficiency.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62272008.

References

Liying Cheng, Xingxuan Li, and Lidong Bing. 2023. Is gpt-4 a good data analyst? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9496–9514.

Shengyue Guan, Haoyi Xiong, Jindong Wang, Jiang Bian, Bin Zhu, and Jian-guang Lou. 2025. Evaluating llm-based agents for multi-turn conversations: A survey. *arXiv preprint arXiv:2503.22458*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Sirui Hong and 1 others. 2023. MetaGPT: Meta programming for a multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.

Xueyu Hu, Ziyu Zhao, Shuang Wei, Ziwei Chai, Qianli Ma, Guoyin Wang, Xuwu Wang, Jing Su, Jingjing Xu, Ming Zhu, Yao Cheng, Jianbo Yuan, Jiwei Li, Kun Kuang, Yang Yang, Hongxia Yang, and Fei Wu. 2024. Infiagent-dabench: evaluating agents on data analysis tasks. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2023. Ds-1000: A natural and reliable benchmark for data science code generation. In *International Conference on Machine Learning*, pages 18319–18345. PMLR.

LangChain. 2024. Pandas dataframe. https://python.langchain.com/v0.2/docs/integrations/toolkits/pandas/. Accessed: 2025-08-04.

Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, Victor Zhong, Caiming Xiong, Ruoxi Sun, Qian Liu, Sida Wang, and Tao Yu. 2025. Spider 2.0: Evaluating language models on real-world enterprise text-to-SQL workflows. In *The Thirteenth International Conference on Learning Representations*.

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 *Advances in Neural Information Processing Systems*.

Zijian Li, Xin Guan, Bo Zhang, Shen Huang, Houquan Zhou, Shaopeng Lai, Ming Yan, Yong Jiang, Pengjun Xie, Fei Huang, and 1 others. 2025. Webweaver: Structuring web-scale evidence with dynamic outlines for open-ended deep research. *arXiv preprint arXiv:2509.13312*.

Jinwei Lu, Yuanfeng Song, Zhiqian Qin, Haodi Zhang, Chen Zhang, and Raymond Chi-Wing Wong. 2025. Bridging the gap: Enabling natural language queries for nosql databases through text-to-nosql translation. *arXiv preprint arXiv:2502.11201*.

Jinwei Lu, Yuanfeng Song, Chen Zhang, and Raymond Chi-Wing Wong. 2026. Multivis-agent: A multi-agent framework with logic rules for reliable and comprehensive cross-modal data visualization. *Proceedings of the ACM on Management of Data*, 4(1 (SIGMOD):1–25.

Pingchuan Ma, Rui Ding, Shuai Wang, Shi Han, and Dongmei Zhang. 2023. InsightPilot: An LLM-empowered automated data exploration system. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 346–352, Singapore. Association for Computational Linguistics.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang.

2024. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, Bangkok, Thailand. Association for Computational Linguistics.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. 2025. Discoverybench: Towards data-driven discovery with large language models. In *International Conference on Learning Representations (ICLR)*.

OpenAI. 2024. Data analysis with chatgpt. https://help.openai.com/en/articles/8437071-data-analysis-with-chatgpt. Accessed: 2025-08-04.

Gaurav Sahu, Abhay Puri, Juan A. Rodriguez, Amirhossein Abaskohi, Mohammad Chegini, Alexandre Drouin, Perouz Taslakian, Valentina Zantedeschi, Alexandre Lacoste, David Vazquez, Nicolas Chapados, Christopher Pal, Sai Rajeswar, and Issam H. Laradji. 2025. Insightbench: Evaluating business analytics agents through multi-step insight generation. In *The Thirteenth International Conference on Learning Representations*.

Yexuan Shi, Mingyu Wang, Yunxiang Cao, Hongjie Lai, Junjian Lan, Xin Han, Yu Wang, Jie Geng, Zhenan Li, Zihao Xia, and 1 others. 2025. Aime: Towards fully-autonomous multi-agent framework. *arXiv preprint arXiv:2507.11988*.

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*.

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*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. 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.

Hui Wei, Zihao Zhang, Shenghua He, Tian Xia, Shijia Pan, and Fei Liu. 2025. Plangenllms: A modern survey of llm planning capabilities. *arXiv preprint arXiv:2502.11221*.

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)*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Zhenghao Zhu, Yuanfeng Song, Xin Chen, Chengzhong Liu, Yakun Cui, Caleb Chen Cao, Sirui Han, and Yike Guo. 2026. Insighteval: An expert-curated benchmark for assessing insight discovery in llm-driven data agents. In *Findings of the Association for Computational Linguistics: ACL 2026*.

A Additional Experimental Details

A.1 Evaluation Metrics

We follow standard practice to evaluate each output twice and take the average score to reduce variance. Following the methodology of InsightBench (Sahu et al., 2025), we adopt the following two evaluation metrics, with detailed definitions provided below. Both metrics are assessed using GPT-4o.

- **Insight-level Scores**: Evaluate individual insights for relevance, novelty, and depth. For each dataset, we generate multiple insights and compute the average G-Eval score across all insights. This metric captures the quality of individual discoveries.

$$\text{Insight-Level Score} = \frac{1}{|GT|} \sum_{gt \in GT} \max_{a \in A} \mathcal{G}(gt, a)$$

  where GT is the set of ground-truth insights, A is the set of agent-generated insights, $|GT|$ denotes the number of ground-truth insights, and $\mathcal{G}$ is the G-Eval evaluator that computes the similarity score between two insights.

- **Summary-level Scores**: Assess the overall coherence, comprehensiveness, and actionability of the final integrated insight summary.

$$\text{Summary-Level Score} = \mathcal{G}(S_{\text{agent}}, S_{\text{gt}})$$

  where S_{agent} is the summary generated by the agent, $S_{\text{ground-truth}}$ is the ground-truth summary, and $\mathcal{G}$ is the G-Eval evaluator that computes the similarity score between two summaries.

A.2 Implementation Details

This section provides comprehensive implementation details for reproducing the experiments.

Model Configuration. The model configurations for DataSeer and all baselines are carefully chosen to ensure fair comparison and reproducibility. Below we specify the key components:

- **Primary LLM:** The core reasoning and code generation for the DataSeer agents and all baselines (ReAct, Pandas Agent, AgentPoirot) is performed using the **DeepSeek-V3-250324** and **gpt-4o-2024-11-20** model via its official API, **Aime** uses its official, default model configuration.
- **Evaluation LLM:** The G-Eval metrics for both insight-level and summary-level scores are computed using **gpt-4o-2024-11-20** as the judge model.
- **Sampling Parameters:** The **temperature** for all LLM calls is set to **0** to ensure deterministic outputs.

Framework Hyperparameters. Key operational parameters of the DataSeer framework are set as follows:

- **Planning-Execution Loop:** The *Planner* generates and refines sub-goals for a maximum of **10 rounds**. For each sub-goal, the *Executor* conducts tactical exploration, also capped at **10 action steps**.
- **Multi-Branch Reasoning (MBR):** The MBR module launches **2** parallel discovery branches. Their final insights are merged by the *Manager* using an LLM-based synthesis prompt.
- **Memory Retrieval:** The top-k value for retrieving relevant memory units from Long-Term Memory (LTM) is set to 5.
- **Quality Threshold:** The Executor's self-evaluation score threshold (θ) for accepting an insight is set to 0.7.

B Additional Experimental Results

B.1 Detailed Ablation Study

To provide a more granular analysis of the contributions of each core component in DataSeer, we conduct fine-grained ablation experiments on InsightBench and InsightEval. Unlike the main experiments that treat the MPE framework as a single module, here we decompose MPE into two distinct subcomponents: **Manager** and **Planner-Executor collaborative loop (Loop)**, and separately examine them alongside **Multi-Branch Reasoning (MBR)**. Results are detailed in Tables 5 and 6.

Overall Trend: The full three-component configuration (✓, ✓, ✓) achieves the highest average insight-level and summary-level scores across all difficulty levels on both benchmarks, validating the effectiveness of the integrated design.

Component Contribution Analysis:

- **Role of Manager:** The Manager, through its dual-layer memory system and dynamic

Components			Insight-level Scores (G-Eval)				Summary-level Scores (G-Eval)			
Manager	Loop	MBR	**Easy**	**Medium**	**Hard**	**Avg**	**Easy**	**Medium**	**Hard**	**Avg**
✓	✓	✓	**0.5852**	**0.4877**	**0.4712**	**0.5117**	**0.6240**	**0.6102**	**0.5803**	**0.6048**
✗	✓	✓	0.5885	0.4723	0.4416	0.4973	0.6186	0.6092	0.5580	0.5956
✗	✗	✓	0.5522	0.4700	0.4077	0.4747	0.5289	0.5102	0.4641	0.5011
✓	✓	✗	0.5269	0.4686	0.4379	0.4762	0.5967	0.5771	0.5570	0.5765
✗	✓	✗	0.5211	0.4538	0.4166	0.4621	0.5709	0.5863	0.5421	0.5675
✗	✗	✗	0.5113	0.4295	0.3579	0.4311	0.4740	0.4476	0.3584	0.4270

Table 5: Detailed Ablation Study of DataSeer on InsightBench

Components			Insight-level Scores (G-Eval)				Summary-level Scores (G-Eval)			
Manager	Loop	MBR	**Easy**	**Medium**	**Hard**	**Avg**	**Easy**	**Medium**	**Hard**	**Avg**
✓	✓	✓	**0.5971**	**0.5455**	**0.5227**	**0.5537**	**0.7187**	**0.6860**	**0.6683**	**0.6901**
✗	✓	✓	0.5911	0.5171	0.5168	0.5506	0.6919	0.6776	0.6473	0.6721
✗	✗	✓	0.5643	0.5170	0.4862	0.5213	0.7029	0.6684	0.6233	0.6643
✓	✓	✗	0.5666	0.5249	0.4947	0.5278	0.6461	0.6491	0.6685	0.6544
✗	✓	✗	0.5635	0.5099	0.4673	0.5124	0.6759	0.6599	0.6207	0.6521
✗	✗	✗	0.5382	0.4948	0.4509	0.4937	0.5762	0.5560	0.5234	0.5517

Table 6: Detailed Ablation Study of DataSeer on InsightEval

prompt editing, primarily ensures cross-episode contextual coherence. Removing the Manager while preserving Loop and MBR leads to a significant drop in summary-level scores on InsightBench (from 0.6048 to 0.5956, -1.5%) and a moderate drop on InsightEval (from 0.6901 to 0.6721, -2.6%). This highlights the Manager's critical role in maintaining narrative coherence and global coordination, particularly in complex, multi-step analytical scenarios.

- **Role of Planner-Executor Loop:** The collaborative loop between Planner and Executor forms the core analytical engine, responsible for hierarchical task decomposition, adaptive planning, and tactical exploration. Removing the Loop (while keeping Manager and MBR) causes substantial performance degradation. On InsightBench, average insight-level and summary-level scores drop by 7.2% and 17.1%, respectively; on InsightEval, they decrease by 5.9% and 3.7%. This confirms that the iterative, goal-driven interaction between Planner and Executor is indispensable for high-quality insight generation.

- **Role of Multi-Branch Reasoning (MBR):** MBR mitigates single-path stochasticity by running multiple discovery trajectories and synthesizing their outcomes. Disabling MBR (with Manager and Loop active) results in moderate declines in both insight-level (InsightBench: -6.9%, InsightEval: -4.7%) and summary-level scores. These results underscore MBR's value in enhancing insight quality through multi-path exploration.

Synergistic Effects: The experiments further reveal synergistic interactions among components. For instance, the configuration without Manager but with Loop and MBR ($\times$, ✓, ✓) still achieves relatively high summary-level scores (0.5956 on InsightBench), suggesting that a strong planning-execution loop combined with multi-branch exploration can partially compensate for the absence of centralized coordination. However, when both Manager and Loop are removed, leaving only MBR ($\times$, $\times$, ✓), performance deteriorates sharply—especially on InsightBench, where summary-level score drops to 0.5011. This indicates that parallel exploration alone, without systematic task decomposition and execution, cannot ensure analytical depth and quality.

Comparison with Baseline: The configuration with all three components disabled ($\times$, $\times$, $\times$) corresponds to the AgentPoirot baseline. Compared to this baseline, the full DataSeer framework achieves improvements of 41.6% in average summary-level score and 18.7% in average insight-level score on InsightBench, and 25.1% and 12.1% on InsightEval, respectively. This substantial gap clearly quantifies the performance gains brought by the proposed manager-centric collaborative framework and multi-branch reasoning mechanism.

In summary, the fine-grained ablation study confirms that the Manager, Planner-Executor Loop, and MBR each play distinct and crucial roles, and

Id	DataSeer				AgentPoirot			
	Coverage	**Reasonableness**	**Avg**	**G-Eval**	**Coverage**	**Reasonableness**	**Avg**	**G-Eval**
1	10.0	10.0	10.0	9.0	3.0	6.0	4.5	3.1
2	4.0	3.0	3.5	5.9	5.0	7.0	6.0	5.2
3	10.0	10.0	10.0	8.8	8.0	8.0	8.0	8.0
4	6.0	8.0	7.0	6.2	6.0	7.0	6.5	7.9
5	9.0	10.0	9.5	8.2	2.0	5.0	3.5	6.0
6	6.0	5.0	5.5	5.0	3.0	4.0	3.5	2.1
7	6.0	7.0	6.5	7.6	2.0	4.0	3.0	2.9
8	4.0	5.0	4.5	5.8	3.0	4.0	3.5	3.0
9	8.0	9.0	8.5	5.2	7.0	9.0	8.0	3.5
10	5.0	7.0	6.0	7.9	5.0	7.0	6.0	3.0
11	7.0	8.0	7.5	7.3	5.0	4.0	4.5	4.2
12	3.0	5.0	4.0	4.8	3.0	2.0	2.5	2.6
13	4.0	8.0	6.0	6.1	2.0	3.0	2.5	2.8
14	6.0	6.0	6.0	5.7	2.0	2.0	2.0	2.1
15	6.0	7.0	6.5	6.0	5.0	7.0	6.0	6.0
16	5.0	7.0	6.0	6.2	3.0	3.0	3.0	3.8
17	8.0	9.0	8.5	8.3	6.0	5.0	5.5	6.0
18	9.0	9.0	9.0	8.0	7.0	7.0	7.0	7.2
19	9.0	9.0	9.0	8.4	5.0	6.0	5.5	6.0
20	9.0	8.0	8.5	7.7	4.0	5.0	4.5	5.6
Avg	**6.9**	**7.5**	**7.2**	**6.8**	**4.3**	**5.3**	**4.8**	**4.7**

Table 7: Comparative Human Evaluation and G-Eval Scores (0-10 scale)

their integration creates synergistic effects that collectively underpin DataSeer's superior performance in automated insight discovery.

B.2 Human Evaluation on Insight Discovery Performance

To complement automated evaluation with human assessment, we invited domain experts to conduct a comparative evaluation of outputs from DataSeer and AgentPoirot on 20 InsightEval tasks. The evaluation dimensions are as follows:

- **Coverage:** Extent of analytical exploration.
- **Reasonableness:** Logical coherence and methodological soundness.

Scores range from 1-10 (10 = excellent). For consistency with human scoring, G-Eval scores (originally 0-1) are scaled to 0-10. Table 7 presents human and G-Eval scores (both on 0-10 scale).

Key Findings:

- **Overall Superiority:** DataSeer outperforms AgentPoirot by 50.0% in human average score (7.2 vs. 4.8) and by 44.7% in G-Eval score (6.8 vs. 4.7).
- **Dimensional Strengths:** DataSeer excels in *Reasonableness* (7.5 vs. 5.3, +41.5%) and *Coverage* (6.9 vs. 4.3, +60.5%).
- **Evaluation Consistency:** Both systems show strong relationship between human and G-Eval scores. The close alignment between human and automated evaluation indicates that G-Eval provides a reliable proxy for human judgment in this task domain.

B.3 Performance Across Diverse Backbone LLMs

To evaluate the robustness of DataSeer across different underlying LLMs, we conduct experiments using **GPT-4o** (`gpt-4o-2024-11-20`) and **DeepSeek-V3** (`DeepSeek-V3-250324`) as backbones. Results on InsightBench and InsightEval are shown in Figures 2 and 3.

InsightBench. DataSeer outperforms all baselines with both LLMs. Using GPT-4o, it achieves the highest average insight-level score (0.5254), surpassing the best GPT-4o baseline (AgentPoirot) by 12.9%, and a strong summary-level score (0.5888). Using DeepSeek-V3, DataSeer achieves the highest average summary-level score (0.6048), exceeding the best baseline (Aime) by 11.6%, and a competitive insight-level score (0.5117). No-

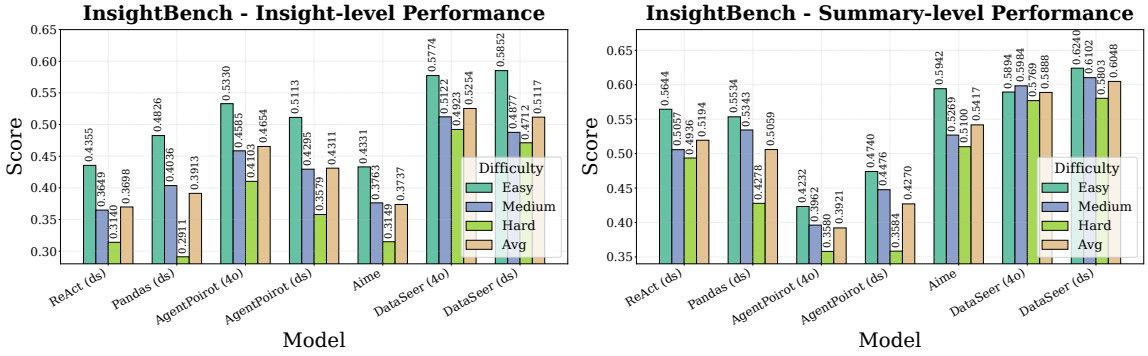

Figure 2: Comparative Performance Analysis of Various Methods on InsightBench

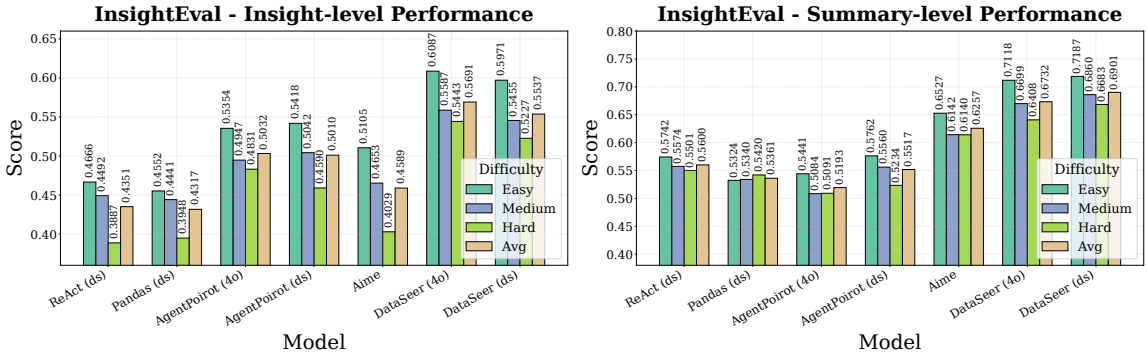

Figure 3: Comparative Performance Analysis of Various Methods on InsightEval

tably, DataSeer with DeepSeek-V3 attains the top summary-level scores across all difficulty levels, demonstrating effective narrative synthesis even with a less powerful backbone.

InsightEval. A similar trend is observed. With GPT-4o, DataSeer leads in average insight-level score (0.5691, +13.1% over AgentPoirot) and summary-level score (0.6732). With DeepSeek-V3, it again achieves the highest average summary-level score (0.6901, +10.3% over Aime) and a solid insight-level score (0.5537). The consistency across difficulty levels confirms the framework's robustness.

Key Observations. Across both datasets, DataSeer consistently ranks at the top in summary-level scores and is highly competitive in insight-level scores, regardless of the backbone LLM. The advantage is more pronounced in summary-level evaluation, underscoring the framework's strength in maintaining narrative coherence.

B.4 Hyperparameter Study

In this subsection, we investigate the sensitivity of DataSeer to two key hyperparameters: the sampling temperature of the backbone LLM and the number of parallel branches in Multi-Branch Reasoning (MBR). Through these studies, we aim to understand their impact on the overall performance and justify our default parameter configurations.

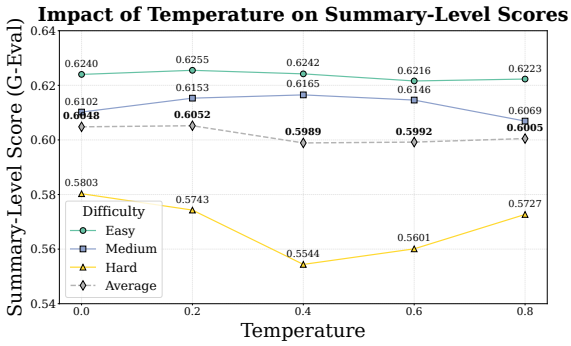

Figure 4: Impact of Model Temperature on the Summary-Level Scores of InsightBench

B.4.1 Behavior of Temperature

We investigate the influence of the temperature hyperparameter on DataSeer's performance using InsightBench (summary-level scores) as shown in Figure 4. Temperature values from 0.0 to 0.8 are tested while keeping all other settings unchanged.

Core Finding: Different temperature settings produce very close performance, with average

scores varying by less than 1.1% (0.5989–0.6052). This indicates that the collaborative framework of DataSeer is robust to sampling randomness and does not rely heavily on stochastic sampling for exploration. The slightly higher scores at temperature = 0.0 and 0.2 suggest that a deterministic or near-deterministic sampling mode best supports the framework's structured reasoning and memory-guided planning.

B.4.2 Scaling Behavior of Multi-Branch Reasoning

We empirically investigate the performance scaling of Multi-Branch Reasoning (MBR) across varying numbers of parallel trajectories ($k \in \{1, 2, 3\}$). Table 8 presents the summary-level scores on InsightEval as the branch count increases.

Core Finding: As shown in Table 8, increasing the number of branches consistently improves summary-level scores across all difficulty subsets, though the marginal gains exhibit diminishing returns. Transitioning from a single trajectory ($k = 1$) to the default dual-branch setup ($k = 2$) yields a substantial performance leap, with the average G-Eval score improving by 0.0357 (from 0.6544 to 0.6901). Further scaling to $k = 3$ achieves the highest overall performance (0.7008), yet the absolute improvement (+0.0107) is markedly smaller than the initial leap.

This saturation effect occurs because, as k increases, the final synthesized output gradually converges toward the full union of all correct insights identifiable by the system. Furthermore, because each branch operates with independent Planner and Executor components while sharing only the centralized Manager, adding more branches reliably broadens insight coverage without introducing cross-branch interference or compromising generation quality. Given the trade-off between these plateauing performance gains and the linear increase in computational overhead (i.e., inference latency and token consumption), we adopt $k = 2$ as the optimal baseline for our standard experimental settings.

B.5 Case Study

To provide a qualitative assessment of DataSeer's analytical capabilities, we conduct a case study comparing its performance against baseline methods across four critical dimensions: data quality sensitivity, actionability, anomaly recognition, and analytical depth.

Branch	Summary-level Scores (G-Eval)			
	Easy	Medium	Hard	Avg
$k = 1$	0.6461	0.6491	0.6685	0.6544
$k = 2$	0.7187	0.6860	0.6683	0.6901
$k = 3$	0.7320	0.6928	0.6813	0.7008

Table 8: Scaling behavior of Multi-Branch Reasoning on InsightEval. Performance consistently improves as the number of parallel branches increases, but marginal gains diminish beyond $k = 2$.

B.5.1 Data Quality and Integrity Awareness

DataSeer demonstrates heightened sensitivity to data quality issues that may compromise analytical validity. As shown in Table 9, while baseline methods often accept surface patterns at face value, DataSeer exhibits statistical skepticism, recognizing improbable uniformity as a potential data artifact. This meta-analytical capability extends to systematic assessment of data completeness and distributional characteristics, preventing erroneous conclusions drawn from incomplete or anomalous data. By integrating data quality evaluation directly into the analytical workflow, DataSeer ensures insights are grounded in reliable evidence.

B.5.2 Emphasis on Actionable Recommendations

A key distinction between DataSeer and baseline approaches lies in its operational orientation. Table 10 illustrates how DataSeer consistently transforms analytical observations into concrete, implementable recommendations. Unlike baseline methods that stop at descriptive statistics, DataSeer develops hierarchical intervention designs with clear accountability, proposes systemic process redesigns based on engineering principles, establishes multi-layered risk mitigation frameworks, and provides strategic procurement optimizations. This focus on actionability ensures analytical insights deliver tangible business value.

B.5.3 Ability to Handle Ambiguity and Recognize Anomalies

DataSeer excels at identifying when apparent normality masks deeper anomalies or when counter-intuitive patterns reveal important insights. As demonstrated in Table 11, DataSeer employs anomaly-driven investigation to detect statistical improbabilities that signal data artifacts, uncovers counter-intuitive relationships that challenge conventional assumptions, and recognizes distributional patterns indicative of underlying process

DataSeer	Baseline Method	Our Advantage
DataSeer identifies a perfectly balanced incident distribution (100 per category) as statistically unusual, suggesting synthetic data. It then examines priority, cost, and resolution time to reveal underlying disparities.	The baseline method notes the "perfectly balanced" distribution as an unremarkable fact and does not pursue further investigation.	*Statistical Skepticism:* Recognizes improbable uniformity as a data artifact and initiates multi-dimensional analysis, uncovering meaningful variations masked by surface balance.
DataSeer uncovers some data quality issues, including missing `processed_date` values and bimodal processing times that indicate workflow bottlenecks.	The baseline method finds a negative correlation between expense amount and processing time but does not examine underlying data completeness or anomalies.	*Integrated Quality Assessment:* Systematically evaluates data completeness and distributional characteristics, preventing conclusions drawn from incomplete or anomalous data.

Table 9: Data Quality and Integrity Sensitivity Comparison

Our Method	Baseline Method	Our Advantage
DataSeer recommends specific, targeted actions: preventive maintenance for Printer546 (Australia), performance-based training, and seasonal resource adjustments. Steps are prioritized.	The baseline method provides only generic observations (e.g., "resolution times vary") without proposing concrete interventions.	*Hierarchical Intervention Design:* Translates analytical observations into prioritized, context-specific action plans with clear accountability and resource optimization.
DataSeer proposes concrete systemic changes: capping individual assignments, reassigning tasks by performance, and investigating root causes.	The baseline method notes workload imbalances but offers no specific redistribution or process-change solutions.	*Systemic Process Redesign:* Provides actionable recommendations for modifying operational rules and resource allocation based on process engineering principles for sustainable improvement.
DataSeer proposes a multi-pronged fix: spot audits, flag round-figure claims, collusion checks, and auto-alerts.	The baseline method observes potential fraud indicators but proposes no audit procedures or controls.	*Risk Mitigation Framework:* Delivers a multi-layered control framework addressing immediate fraud risks while strengthening the organizational control environment.
DataSeer advocates structural improvements: standardizing policies, renegotiating contracts, and investigating vendor anomalies.	The baseline method compares costs and vendor variations without suggesting specific negotiation strategies or policy changes.	*Strategic Procurement Optimization:* Generates insights for policy standardization, targeted contract renegotiation, and data-driven vendor management, moving beyond cost comparisons.

Table 10: Actionability and Implementation Focus Comparison

variations. This capability prevents superficial acceptance of misleading patterns and enables discovery of non-obvious relationships that baseline methods often overlook.

B.5.4 Depth of Analysis and Insight Generation

The most significant advantage of DataSeer lies in its analytical depth. As Table 12 shows, DataSeer consistently moves beyond surface observations to uncover root causes, systemic relationships, and nuanced patterns through multi-dimensional, hypothesis-driven analysis. It performs comprehensive causal analysis that quantifies imbalances and identifies systemic origins, employs granular hypothesis testing to avoid oversimplified conclusions, conducts multi-dimensional diagnostics to precisely identify interacting factors, provides strategic governance assessments beyond operational efficiency, and detects systemic risks missed by individual case review. This transformative insight generation capability enables organizations to address not just symptoms but underlying causes.

B.5.5 Synthesis of Analytical Advantages

The systematic comparison across these four critical dimensions reveals a consistent pattern of analytical superiority in DataSeer. Unlike the baseline approach that primarily offers descriptive observations, DataSeer demonstrates four key analytical advantages:

1. **Multi-dimensional analytical frameworks** that integrate quantitative metrics, qualitative patterns, temporal trends, and contextual factors into cohesive assessments across all analytical dimensions.

2. **Hypothesis-driven investigation** that systematically tests assumptions and explores alternative explanations rather than accepting surface patterns, particularly evident in anomaly recognition and depth of analysis.

3. **Meta-analytical capabilities** that evaluate data quality and methodological appropriateness as integral components of the analytical process, ensuring insights are grounded in reliable evidence.

4. **Transformative insight generation** that moves beyond describing "what" happened to explaining "why" it happened and prescribing "how" to address it, bridging the gap between analysis and actionable implementation.

These capabilities collectively enable DataSeer to deliver actionable, reliable, and profound insights that drive meaningful organizational improvements, addressing limitations in conventional analytical approaches.

B.5.6 Complementary Insight Synthesis through Multi-Agent Collaboration

Beyond the core analytical capabilities, DataSeer demonstrates a significant structural advantage through its multi-agent parallel trajectory design. A recurring pattern in

Our Method	Baseline Method	Our Advantage
DataSeer identifies a perfectly balanced incident distribution as statistically improbable, prompting investigation of other dimensions. It discovers significant disparities in priority, resolution efficiency, and user satisfaction masked by the artificial balance.	Accepts balanced distribution as unremarkable, concluding that "no single category requires additional focus based on volume alone." This demonstrates superficial acceptance without critical examination.	*Anomaly-Driven Investigation:* Detects statistical improbabilities that signal data artifacts, prompting multi-dimensional analysis to uncover meaningful variations and prevent Type II errors.
DataSeer discovers that manually updated tickets are resolved faster than system-updated ones, a counter-intuitive finding that challenges assumptions about automation efficiency. This prompts investigation of potential workflow bottlenecks.	Observes category-level TTR variations but fails to examine differences in update method efficiency, missing opportunities to identify automation inefficiencies.	*Counter-Intuitive Insight:* Identifies and investigates findings that contradict conventional wisdom, revealing hidden optimization opportunities and challenging organizational assumptions.
DataSeer identifies bimodal peaks in expense processing times, suggesting distinct workflow paths or approval bottlenecks rather than a unimodal distribution. It recognizes that the relationship between expense amount and processing time exhibits threshold effects and process variations, not just linear correlation.	Notes a negative correlation between expense amount and processing time but misses distributional anomalies and threshold effects. Analysis remains at aggregate correlation level without examining underlying process variations.	*Distributional Pattern Recognition:* Detects multi-modal distributions and threshold effects indicative of underlying process variations, enabling targeted improvements rather than blanket efficiency measures.

Table 11: Anomaly Recognition and Ambiguity Handling Comparison

Our Method	Baseline Method	Our Advantage
DataSeer quantifies workload imbalance precisely (77.15% high-priority tasks), identifies automation bias as root cause (73.29% system-assigned), links high workload to efficiency degradation, and proposes systemic solutions addressing both symptoms and causes.	Notices aggregate workload disparities without quantifying specific imbalances, identifies no systemic causes, and proposes no solutions connecting patterns to underlying processes.	*Comprehensive Causal Analysis:* Quantifies specific imbalances, identifies systemic root causes, examines operational consequences, and designs interventions addressing both immediate symptoms and underlying drivers for sustainable improvement.
DataSeer confirms superior departmental performance while revealing priority-level variations: critical/high-priority goals consistently overachieve while medium-priority goals severely underperform. Systematically tests and rejects tenure as an explanatory factor.	Identifies aggregate performance differences without examining priority-level variations or testing alternative explanatory factors. Misses that excellence is priority-specific rather than uniform.	*Granular Hypothesis Testing:* Disaggregates performance across dimensions and systematically tests alternative hypotheses, avoiding oversimplified conclusions and enabling precise, priority-specific interventions.
DataSeer identifies priority misalignment (critical incidents take longer than high-priority ones), discovers server issues as dominant technical cause, quantifies regional disparities (65% longer in Australia), and links delays to descriptive patterns like "slow" and "delay".	Notes aggregate category differences without examining priority alignment, technical root causes, geographical variations, or descriptive patterns. Lacks diagnostic depth.	*Multi-Dimensional Diagnostics:* Simultaneously examines problems across priority alignment, technical causation, geographical distribution, and descriptive language to precisely identify interacting contributing factors.
DataSeer uncovers policy inconsistency through non-standard warranty durations, identifies departmental inequities, detects temporal policy shifts, and assesses coverage risks for critical infrastructure through integrated analysis.	Compares surface metrics like cost per warranty day without examining policy consistency, fairness, temporal trends, or risk exposure. Lacks broader organizational assessment.	*Strategic Governance Assessment:* Evaluates organizational practices across policy consistency, equity, evolution, and risk exposure, providing insights relevant to governance, compliance, and strategic planning beyond operational efficiency.
DataSeer uncovers departmental anomalies and potential collusion indicators through clustering analysis of claim amounts and submitter relationships, detecting systematic patterns that individual anomaly detection would miss.	Focuses only on individual duplicate claims without detecting systemic patterns or potential collusion. Remains at individual transgression level.	*Systemic Risk Detection:* Identifies systematic patterns and potential collusion through clustering and anomaly analysis at the departmental and group levels, revealing systemic control weaknesses missed by individual case review.

Table 12: Analytical Depth and Insight Generation Comparison

our qualitative assessment reveals that individual analytical trajectories, even those generated by advanced LLMs, often exhibit tunnel vision by focusing on specific dimensions while inadvertently ignoring others. By merging parallel analytical pathways, DataSeer consistently synthesizes a comprehensive final summary that captures insights missed by individual agents.

As shown in Table 13, when single trajectories fail to extract specific ground-truth claims due to hypothesis fixation or limited contextual breadth, the final synthesis effectively reconciles these fragmented perspectives. This architecture mitigates the inherent volatility and bias of single-path generation, ensuring that critical findings—such as specific temporal anomalies, subtle distributional shifts, or regional disparities—are not lost. This complementary synthesis enables DataSeer to achieve a higher recall of actionable insights than what any single analytical pass could accomplish alone.

Ground Truth Claim	Trajectory 1 (Partial Focus)	Trajectory 2 (Alternative Focus)	DataSeer Final Synthesis (Combined Success)
"Uniform trend of TTR for all category incidents, however there is a dense cluster of incidents in the Hardware category during the period 2023-08."	Identifies temporal anomalies ("peaks in January and November 2023") and resolution time differences but misses the specific uniform trend context and the dense cluster comparison.	Focuses on data quality issues ("invalid timestamps") and regional disparities, completely missing the temporal trend of TTR clustering.	**Synthesized:** Successfully integrates the temporal trend and the distribution of resolution times alongside data anomalies, capturing both the timeline issues and the specific hardware category clustering patterns missed by individual paths.
"Fluctuations observed in incident frequencies across categories and Increased Hardware incidents from 2023-06 to 2023-08."	Highlights statistical correlations and workflow delays ("slope = 8.92", "mean: 1293.85") but fails to identify the specific temporal window and cross-category fluctuations.	Detects the sharp upward trend ("increasing from 178.23 hours... spike starting July 2023") but lacks the broader cross-category context.	**Synthesized:** Combines the statistical severity from Trajectory 1 with the precise temporal window and categorical context from Trajectory 2, providing a complete picture of the hardware incident surge relative to other categories.
"Hardware incidents predominantly occur in Australia during spikes from 2023-06 to 2023-08."	Identifies Australia as having higher resolution times ("median: 1546.8 hours") but does not link it to the specific temporal spike window.	Identifies the temporal spike ("starting July 2023") and mentions Australia's slow resolution, but fails to explicitly correlate the incident volume predominance in Australia during that specific timeframe.	**Synthesized:** Successfully cross-references the geographical disparity (Australia) with the specific temporal spike (mid-2023), recognizing that the spike is not just a global anomaly but geographically concentrated.

Table 13: Complementary Insight Synthesis via Parallel Trajectories. The final summary successfully captures ground truth claims that were missed or only partially identified by individual analytical agents.

C Prompts for Agents

This appendix provides the key prompt templates used by the agents in the DataSeer framework. Prompt 1 and 2 present the detailed user prompts and system prompts for agents in DataSeer.

Prompt 1: User message for Planner thought generation.

```
You are a data analysis expert who is improving your analytical thinking based on
    reflection results.

Given the following goal:
<goal>{goal}</goal>

Given the following data schema:
<schema>{schema}</schema>

Given the following history:
<history>{history}</history>

Given the following reflection results:
<reflection>{reflection}</reflection>

You may also refer to the following related memories (examples of past experiences,
    lessons learned, or recurring patterns).
<related_memories>{additional_context}</related_memories>

### Important Note:
- Do NOT treat the related memories as facts about the current task.
- Instead, extract useful patterns, strategies, and lessons.
- Focus on generalizable insights rather than repeating or inventing specific
    details from the memories.

### Instructions:
You are now at a decision point in your analysis.

Based on the above reflection results, generate new analytical thinking. Please
    think step by step about the current status of the goal:
- What subgoals or aspects of the goal have already been addressed?
- What remains unexplored or under-analyzed?
- Are there questions in the history that are incomplete or worth refining?
- What type of action would help advance the analysis now (e.g., exploring the goal,
     formulating questions, answering, summarizing)?
- Ensure that the new thinking can more effectively advance the analysis objective

Use your thought to reflect on what should be done next. The thought should:
* Assess what has been achieved so far,
* Identify current bottlenecks or missing information,
* Propose a logical next direction in the analysis.

Output your reasoning in the following tag:
<thought>...</thought>

### Response:
```

Prompt 2: System message for Planner thought generation.

```
You are a self-improving data analysis expert.
You excel at adjusting your analytical strategies based on reflection and criticism,
     continuously improving analysis quality.
Your thinking process should:
- Fully absorb suggestions from reflection
- Critically evaluate previous analytical methods
- Flexibly adjust analytical direction and strategy
- Maintain focus on the analysis objective

Your goal is to generate higher quality, more insightful analysis results through
    continuous reflection and improvement.
```

Prompt 3 and 4 guide the conversion of analytical thoughts into concrete sub-goals and execution plans.

Prompt 3: User message for plan generation.

```
You are a strategic planner for data analysis tasks.

<goal>{goal}</goal>

The dataset has the following schema:
<schema>{schema}</schema>

History of previous steps:
<history>{history}</history>

Your current thought:
<thought>{thought}</thought>

You may also refer to the following related memories (examples of past experiences,
    lessons learned, or recurring patterns).
<related_memories>{additional_context}</related_memories>

### Important Note:
- Do NOT treat the related memories as facts about the current task.
- Instead, extract useful patterns, strategies, and lessons.
- Focus on generalizable insights rather than repeating or inventing specific
    details from the memories.

### Instructions:
Based on your thought, create a concrete plan for the next step in the analysis.
Your plan should:
* Be specific and actionable
* Focus on a single, well-defined task
* Consider the available data and schema
* Build upon previous steps in the history

When creating your plan, consider:
* What specific question needs to be answered next?
* What data exploration or analysis is needed?
* If the history contains results from previous plans, evaluate these results and
    adjust your plan accordingly

### Output Format:
* The first line of the output should be the goal of the plan

Output your plan in the following tag:
<plan>...</plan>

### Response:
```

Prompt 4: System message for plan generation.

```
You are an expert data analyst specializing in creating clear, actionable summaries.
Your summaries distill complex analyses into key insights and recommendations.
You focus on addressing the original goal and providing value to decision-makers.
```

Prompt 5 and 6 guide the Planner's reflection process.

Prompt 5: User message for Planner reflection.

```
You are a professional data analysis reflection expert. Your task is to critically
    reflect on the previous analysis process to improve the quality of subsequent
    analysis.

Given the following goal:
<goal>{goal}</goal>

Given the following data schema:
<schema>{schema}</schema>

Given the following history:
<history>{history}</history>

You may also refer to the following related memories (examples of past experiences,
    lessons learned, or recurring patterns).
<related_memories>{additional_context}</related_memories>

### Important Note:
- Do NOT treat the related memories as facts about the current task.
- Instead, extract useful patterns, strategies, and lessons.
- Focus on generalizable insights rather than repeating or inventing specific
    details from the memories.

### Instructions:
Critically reflect on the previous analysis process. Your reflection should:
1. Evaluate the effectiveness and quality of the previous analysis
2. Recognize important aspects or angles that may have been overlooked
3. Propose improvement suggestions to enhance the quality of subsequent analysis

In your reflection, consider the following questions:
- Did the previous analysis effectively advance the objective?
- Are there better methods to solve the problem?
- Are there important data or relationships that were overlooked?
- Are there logical errors or unreasonable assumptions?

Your answer should be concise and to the point.

Make sure to output your reflection in the following tag:
<reflection>...</reflection>

### Response:
```

Prompt 6: System message for Planner reflection.

```
You are a rigorous and critical data analysis reflection expert.
Your expertise lies in evaluating the quality of data analysis processes and
    providing constructive criticism and improvement suggestions.
Your reflection should:
- Be based on facts and logic, not subjective judgment
- Provide relatively specific, actionable improvement suggestions
- Consider the overall objective and context of the analysis

Your goal is to help analysts continuously improve their analysis methods and
    thinking, thereby gaining more valuable insights.
```

Prompt 7 and 8 guide the Executor's tactical thought generation.

Prompt 7: User message for Executor thought generation.

```
You are performing one step of an overall analysis pipeline.
Final objective: <final_goal>{final_goal}</final_goal>
Your immediate task (a specific analytical goal for this step): <goal>{goal}</goal>

The dataset has the following schema:
<schema>{schema}</schema>

You have been given the following plan to execute:
<plan>{plan}</plan>

You may also refer to the following related memories (examples of past experiences,
    lessons learned, or recurring patterns).
<related_memories>{additional_context}</related_memories>

### Important Note:
- Do NOT treat the related memories as facts about the current task.
- Instead, extract useful patterns, strategies, and lessons.
- Focus on generalizable insights rather than repeating or inventing specific
    details from the memories.

### Instructions:
Based on the given plan, generate a thought that will guide your next action.
Your thought should:
* Reflect on what the plan is asking you to do
* Consider how to best execute the plan using the available data
* Identify any potential challenges or considerations

Output your reasoning in the following tag:
<thought>...</thought>

### Response:
```

Prompt 8: System message for Executor thought generation.

```
You are a careful and methodical data analyst.
Your job is to think step-by-step about how to execute the given plan using the
    available data.
You reason like a human analyst: prioritizing relevance, avoiding redundancy, and
    focusing on the specific task at hand.
```

Prompt 9, 10 and 11 guide the Executor's action selection.

Prompt 9: User message for action selection.

```
You are now acting based on your reasoning.

You are performing one step of an overall analysis pipeline.
Final objective: <final_goal>{final_goal}</final_goal>
Your immediate task (a specific analytical goal for this step): <goal>{goal}</goal>

Given the following schema:
<schema>{schema}</schema>

Given the following history:
<history>{history}</history>

Your thought:
<thought>{thought}</thought>

### Instructions:
* Now, based on the thought above, select the next action from the available action
    space.
* The action must be output in the following format:
  <action>...</action>
  <additional_info>...</additional_info>

### Action Space and Output Format:
* generate_question_from_goal
    - Description: Formulate a new question directly based on the original goal.
    - Output format:
        <action>generate_question_from_goal</action>

* generate_question_from_question
    - Description: Generate a follow-up question based on a previous question.
    - Additional info:
        - You must output the base question and the answer in the additional_info
    tag.
        - The base question and the answer should be in the following format:
            <base_question>...</base_question>
            <answer>...</answer>
    - Output format:
        <action>generate_question_from_question</action>
        <additional_info>
            <base_question>...</base_question>
            <answer>...</answer>
        </additional_info>

* select_question
    - Description: Select the most promising question to answer next.
    - Additional info:
        - You must output a list of candidate questions in the additional_info tag.
        - The candidate questions should be in the following format:
            <question_list>
                <question>...</question>
                <question>...</question>
            </question_list>
    - Output format:
        <action>select_question</action>
        <additional_info>
            <question_list>
                <question>...</question>
                <question>...</question>
            </question_list>
        </additional_info>
```

Prompt 10: User message for action selection 2.

```
* answer_question
    - Description: Answer the selected question using the dataset.
    - Additional info:
        - You must output the question to be answered in the additional_info tag.
        - The question and the answer should be in the following format:
            <question>...</question>
    - Output format:
        <action>answer_question</action>
        <additional_info>
            <question>...</question>
        </additional_info>

* summarize
    - Description: Summarize the key findings from the given insights.
    - Additional info:
        - You must include the insights to be summarized, which can be raw or
    actionable insights.
        - Format:
            <insights_list>
                <insights>...</insights>
                <insights>...</insights>
                ...
            </insights_list>
    - Output format:
        <action>summarize</action>
        <additional_info>
            <insights_list>
                <insights>...</insights>
                <insights>...</insights>
                ...
            </insights_list>
        </additional_info>

* halt
    - Description: Stop the analysis if the goal has been sufficiently addressed.
    - Output format:
        <action>halt</action>

### Response:
```

Prompt 11: System message for action selection.

```
You are a methodical, step-by-step data analysis agent.
You are reasoning over a dataset in order to fulfill a specific analytical goal.
You can take one of the following actions:
- generate_question_from_goal: pose a list of new questions directly related to the
    analytical goal
- generate_question_from_question: pose a follow-up new question based on a previous
     question
- select_question: choose the most promising question from history to be answered
    next, and provide the list of candidates considered
- answer_question: answer the selected question using the dataset; clearly specify
    which question is being answered
- summarize: summarize the key findings from the given insights
- halt: decide that the analysis is complete
Each action should move the analysis toward meaningful insight aligned with the user
    's goal.
```

Prompt 12 and 13 guide the Executor's quality assessment for insights.

Prompt 12: User message for self-evaluation.

```
You are a seasoned data analyst tasked with evaluating how well the current data
    analysis task **Objective** has been completed.

Plan:
{plan}

History:
{history_summary}

Current answer: {current_answer}

### Evaluation Criteria:
- Score range: 0~10
- 10: Fully achieved the plan objective; delivered deep and comprehensive analysis
- 8~9: Mostly achieved the plan objective; good analytical depth
- 6~7: Partially achieved the plan objective; some analysis but lacks depth
- 4~5: Initial analysis of the plan objective; superficial insights
- 2~3: Started analyzing the goal; minimal progress
- 0~1: Almost no progress or off-target

### Evaluation Dimensions:
1. Relevance to Plan Objective: Does the analysis directly address the plan
    objective?
2. Depth of Analysis: Are valuable insights provided?
3. Completeness: Are the main aspects of the plan covered?
4. Actionability: Are the results practically useful?

Based on the above criteria, output a score from 0~10, enclosed in <score>...</score
    > tags.
Only output the score, no other text.

### Response:
```

Prompt 13: System message for self-evaluation.

```
You are a professional task-completion evaluator. Your responsibilities are:
- Objectively assess the degree of completion of a data-analysis task
- Make a holistic judgment based on the plan, history, and current result
- Output a score from 0~10, with one decimal place
```

Prompt 14 guides the compression of analytical outputs into structured memory units.

Prompt 14: Prompt for memory compression.

```
You are an expert in knowledge abstraction and generalization.

Your task is to abstract specific details from content while preserving its core
    insights, patterns, and principles. The goal is to create a generalized
    representation that can be applied to different contexts without leaking
    specific details of the original sample.

<Content Type>{content_type}</Content Type>
<Abstraction Level>{abstraction_level}</Abstraction Level>
<Original Content>{content}</Original Content>

Abstraction level description:
- Level 1 (Light Abstraction): Preserve most details but remove sample-specific
    identifiers, names, and unique values
- Level 2 (Medium Abstraction): Extract key patterns, methods, and insights while
    generalizing specific details
- Level 3 (High Abstraction): Only distill core principles, strategies, and high-
    level experiences

Please create an abstracted version of the content according to the specified
    abstraction level.

Please use the following format:
<Abstracted Content>Your abstracted content</Abstracted Content>
```

Prompt 15 guides the consolidation of short-term memory into long-term memory.

Prompt 15: Prompt for memory consolidation.

```
You are an expert in knowledge integration and synthesis.

Your task is to consolidate multiple related memories into a coherent memory that
    captures the essential information from all input memories while eliminating
    redundancy.

<Memory List>
{memories}
</Memory List>

Please consolidate these memories by:
1. Identifying common themes, patterns, and principles
2. Removing redundant information
3. Preserving unique insights from each memory
4. Creating a coherent narrative of integrated knowledge

Please use the following format:
<Consolidated Memory>Your consolidated memory</Consolidated Memory>
```

Prompt 16 guides the LLM-based synthesis of insights from multiple discovery branches.

Prompt 16: Prompt for LLM-based merging of multi-branch results.

```
''Output format:
- Use a sober analyst tone; avoid hype. Make it clear to general readers.
- Use {summary1} as the base content foundation
- Supplement with content from {summary2} that is not already covered in {summary1}
- Then include the heading **Key Insights** followed by a deduplicated bullet list
    using "- " bullets. Preserve important numbers and units; do not alter numeric
    values.
- Then include the heading **Actionable Recommendations** followed by a deduplicated
     bullet list using "- " bullets. Recommendations must be practical and specific.
- Reconcile conflicts between the inputs. If numbers or facts differ, prefer
    conservative wording and explain briefly where appropriate.
- Avoid hallucinations. Do not introduce facts that are not supported by the inputs.
- Keep bullets concise; avoid repeating the same point in different words.
- Use exactly these headings: **Key Insights** and **Actionable Recommendations**.
Summary A:
```
{summary1}
```
```

Prompt 17 and 18 guide the generation of the insight summary.

Prompt 17: User message for summary generation.

```
You are tasked with summarizing the results of a data analysis process.

<goal>{goal}</goal>

The dataset has the following schema:
<schema>{schema}</schema>

History of analysis steps:
<history>{history}</history>

You may also refer to the following related memories (examples of past experiences,
    lessons learned, or recurring patterns).
<related_memories>{additional_context}</related_memories>

### Important Note:
- Do NOT treat the related memories as facts about the current task.
- Instead, extract useful patterns, strategies, and lessons.
- Focus on generalizable insights rather than repeating or inventing specific
    details from the memories.

### Instructions:
Create a comprehensive summary of the analysis results.
Your summary should:
* Directly address the original goal
* Highlight the key insights discovered
    - the key insights should be after the tag "**Key Insights**"
    - each key insight should be output in a single line
* Present actionable recommendations
    - the actionable recommendations should be after the tag "**Actionable
    Recommendations**"
* Be concise yet thorough

Output your summary in the following tag:
<summary>...</summary>

### Response:
```

Prompt 18: System message for summary generation.

```
You are tasked with summarizing the results of a data analysis process.

<goal>{goal}</goal>

The dataset has the following schema:
<schema>{schema}</schema>

History of analysis steps:
<history>{history}</history>

### Instructions:
Create a comprehensive summary of the analysis results.
Your summary should:
* Directly address the original goal
* Highlight the key insights discovered
    - the key insights should be after the tag "**Key Insights**"
    - each key insight should be output in a single line
* Present actionable recommendations
    - the actionable recommendations should be after the tag "**Actionable
    Recommendations**"
* Be concise yet thorough

Output your summary in the following tag:
<summary>...</summary>

### Response:
```

D Assistant of AI

During the preparation of this work, the authors used AI to assist with language revision and code optimization. The authors have reviewed and edited all content and take full responsibility for the final manuscript.