Table-as-Search: Formulate Long-Horizon Agentic Information Seeking as Table Completion

Tian Lan, Felix Henry, Bin Zhu, Qianghuai Jia, Junyang Ren
Qihang Pu, Haijun Li, Longyue Wang*, Zhao Xu, Weihua Luo
Alibaba International Digital Commerce

Abstract

Current Information Seeking (InfoSeeking) agents struggle to maintain focus and coherence during long-horizon exploration, as tracking search states, including planning procedure and massive search results, within one plain-text context is inherently fragile. To address this, we introduce **Table-as-Search (TaS)**, a structured planning framework that reformulates the InfoSeeking task as a Table Completion task. TaS maps each query into a structured table schema maintained in an external database, where rows represent search candidates and columns denote constraints or required information. This table precisely manages the search states: filled cells strictly record the history and search results, while empty cells serve as an explicit search plan. Crucially, TaS unifies three distinct InfoSeeking tasks: Deep Search, Wide Search, and the challenging DeepWide Search. Extensive experiments demonstrate that TaS significantly outperforms numerous state-of-the-art baselines across three kinds of benchmarks, including multi-agent framework and commercial systems. Furthermore, our analysis validates the TaS's superior robustness in long-horizon InfoSeeking, alongside its efficiency, scalability and flexibility. Code and datasets are publicly released at `https://github.com/AIDC-AI/Marco-Search-Agent`.

1 Introduction

Information retrieval is undergoing a paradigm shift from simple fact retrieval to complex long-horizon Agentic InfoSeeking (Li et al., 2025b; Team et al., 2025c; Li et al., 2025a; Yao et al., 2022). It necessitates agents to navigate massive web environments and synthesize answers through multi-step reasoning (Li et al., 2025b; Team et al., 2025c; Li et al., 2025a). Mastering this capability is central to next-generation Deep Research Systems (Google, 2025; Team et al., 2025c).

While Large Language Model (LLM)-based agents have emerged as the dominant solution for this task (Team et al., 2025c,b), current paradigms, such as ReAct (Yao et al., 2022), rely heavily on unstructured plain text to manage the search states, including planning procedure and massive search results, which is inherently fragile. Although recent advancements in context management (Wu et al., 2025; Li et al., 2025b) and procedural planning (Prasad et al., 2024; Yu et al., 2025) attempt to mitigate this overhead, they still burden the finite unstructured agent context with tracking massive search states of long-horizon InfoSeeking. Consequently, as the horizon expands, these methods expose agents to the "lost in the middle" (Zhang et al., 2024) phenomenon, leading to error propagation and ineffective exploration (Chen et al., 2025; Tao et al., 2025). For instance, tracking thousands of search results and corresponding planning process in WideSearch (Wong et al., 2025) within a single plain-text trajectory inevitably leads to severe hallucinations and loss of state fidelity.

To address this, we introduce **Table-as-Search (TaS)**, a structured planning framework that reformulates the InfoSeeking as a **Table Completion** task. As illustrated in Figure 1, rather than treating InfoSeeking as unstructured text generation, TaS explicitly maps the user query into a structured schema where rows represent candidate entities and columns denote specific constraints or required information. This table precisely manages the search states: filled cells represent the search history and results, while empty cells serve as pending actions (i.e., explicit search plan). Moreover, by offloading the massive search results to an external database, TaS alleviates the agent's memory burden, preserving the valuable context window for complex reasoning. Specifically, we implement TaS via a multi-agent system centered around a shared database

* Corresponding author: wanglongyue.wly@alibaba-inc.com

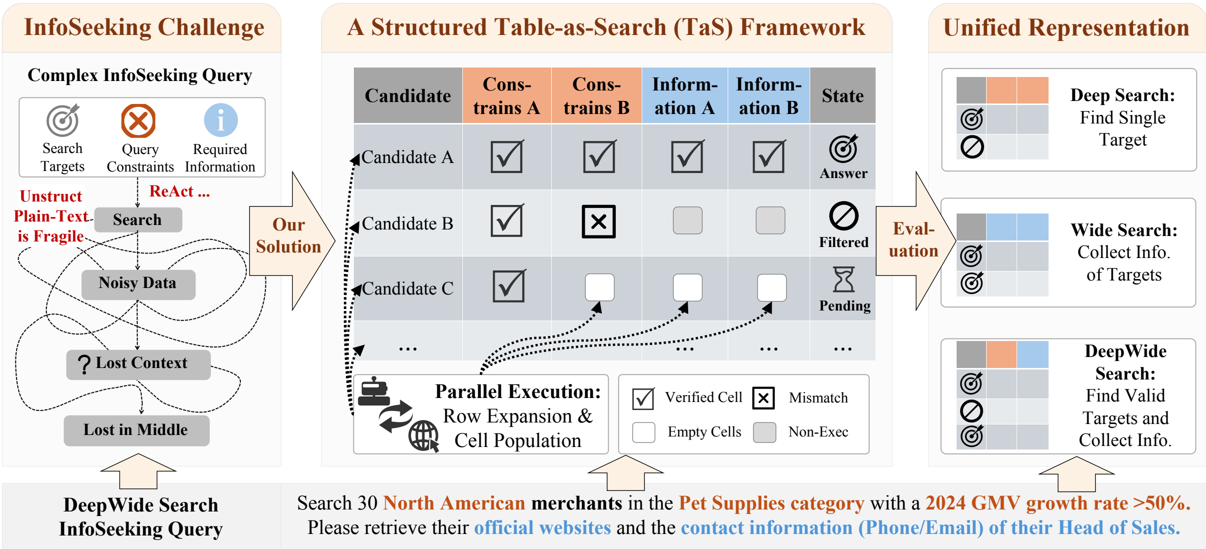

Figure 1: The overview of TaS Framework. **Left:** Unstructured planning (e.g., ReAct) is fragile and prone to massive context. **Center:** TaS reformulates InfoSeeking as Table Completion via row expansion and cell population. **Right:** TaS provides a unified representation for conducting Deep Search, Wide Search and DeepWide Search.

table. A central planner orchestrates sub-agents to iteratively expand rows for candidate discovery and populate cells for constraints verification or information collection.

TaS provides a unified representation for three distinct long-horizon InfoSeeking paradigms: (1) Deep Search: precise target filtering (Wei et al., 2025); (2) Wide Search: broad information aggregation (Wong et al., 2025); and (3) the challenging DeepWide Search (Parallel AI Team, 2025): broad exploration and deep verification. Extensive experiments demonstrate that TaS significantly outperforms state-of-the-art baselines (Yao et al., 2022; Wong et al., 2025; Zhu et al., 2025) across these three kinds of benchmarks. For example, on benchmarks demanding massive search (WideSearch and DeepWide), TaS instantiated with the `Claude-Sonnet-4 (No Think)` significantly outperforms both the computation-heavy Multi-Agent baseline (`Claude-Sonnet-4 (Thinking)`) and the commercial Gemini DeepResearch system. Analysis further highlights TaS's superior robustness as InfoSeeking task complexity increases, alongside its efficiency (higher performance with comparable or lower search volume), scalability (effective test-time scaling), and flexibility (seamless integration of specialized deep search agents).

2 Related Work

Agentic Information Seeking. Recent research categorizes agentic information seeking into three paradigms (Lan et al., 2025): Deep Search (multi-step reasoning for single targets) (Mialon et al., 2023; Wei et al., 2025; Zhou et al., 2025), Wide Search (broad aggregation across extensive sources) (Wong et al., 2025; He et al., 2025), and the hybrid DeepWide Search (Parallel AI Team, 2025). While benchmarks exist for the former two (e.g., BrowseComp (Wei et al., 2025), WideSearch (Wong et al., 2025)), the community lacks public high-quality evaluations for DeepWide InfoSeeking. Addressing this gap, we curate a challenging E-commerce Business Development (BD) benchmark, explicitly designed to stress-test agents in real-world DeepWide InfoSeeking.

Agent Frameworks. The ReAct paradigm (Yao et al., 2022; Liu et al., 2025) serves as the cornerstone of current agentic systems. While recent works have improved ReAct via procedural planning, like Routine (Zeng et al., 2025), ADaPT (Prasad et al., 2024), ReCode (Yu et al., 2025) and ReCAP (Zhang et al., 2025). However, these methods still remains bound by unstructured plain-text planning, facing the same problem of ReAct in long-horizon InfoSeeking. Justified by this shared limitation, we employ the state-of-the-art Multi-Agent ReAct framework (Wong et al., 2025; Kim et al., 2025) as the representative baseline for these unstructured approaches. In contrast, TaS is orthogonal to these methods, introducing a data-centric structure to manage massive search states.

Context Management. To mitigate context overflow, recent approaches employ strategies like context summarization (Wu et al., 2025), folding (Ye et al., 2025) or multi-agent context isolation (Wong et al., 2025). However, they still suffer from lossy compression and the imprecise unstructured recording of search states. In contrast, TaS is orthogonal to these strategies; rather than compressing text, it imposes a structured schema on the search process. Crucially, while TaS can seamlessly incorporate these strategies (as demonstrated in Section 5.3), its distinct advantage lies in offloading massive search results to a structured external database for on-demand access, reserving agent's reasoning capacity for complex decision-making rather than passive information storage.

3 Task Formulation

3.1 Problem Definition

Formally, an InfoSeeking task is defined as a tuple $\mathcal{T} = \langle q, \mathcal{W} \rangle$, where an agent interacts with the web environment $\mathcal{W}$ to fulfill a complex query q. The interaction unfolds over T steps, generating a trajectory (history) $\tau_T = (o_1, r_1, a_1, \ldots, o_T, r_T, a_T)$, where o_t, r_t and a_t denote observations, chain-of-thoughts and actions, respectively (Fang et al., 2025). Standard paradigms (e.g., ReAct) model the agent's policy π as generating the next action conditioned on the entire unstructured history τ_t: $r_{t+1}, a_{t+1} \sim \pi(\cdot \mid q, \tau_t)$. Critically, as the horizon t extend, the relevant information density in τ_t dilutes, causing the "lost-in-the-middle" phenomenon (Chen et al., 2025). The agent must implicitly perform information extraction and state tracking simultaneously within a single forward pass. This challenges agents to propose plans for effective exploration in the search space.

3.2 Table-as-Search (TaS) Framework

To resolve this, we reformulate the InfoSeeking task as a Table Completion problem for precise search state management.

Structured Schema Definition. Instead of operating on free-form text, we map the query q into a structured schema $\mathcal{S}$: $\phi(q) \to \mathcal{S}$. The schema is defined as a tuple of attribute sets: $\mathcal{S} = \langle \mathcal{K}, \mathcal{C}, \mathcal{I} \rangle$. $\mathcal{K}$ uniquely represents the key candidates, $\mathcal{C}$ denotes the Constraint Set, and $\mathcal{I}$ denotes the Information Set (information to be collected). This formulation generalizes to distinct InfoSeeking paradigms by simply varying the set configurations.

Search as Table Completion. As shown in Figure 1, we maintain the long-horizon InfoSeeking as a table T_t, where rows correspond to discovered or potential candidates $e \in \mathcal{E}$ and columns correspond to the schema $\mathcal{S}$. Let $T_t[i, j]$ denote the cell for the i-th candidate and j-th attribute. The cell takes values from $\mathcal{V} \cup \{\emptyset, \text{N/A}\}$, where $\emptyset$ represents a "pending" state and N/A denotes the information that do not need to retrieve. Under this formulation, the policy π is conditioned on a structured table and trajectory: $r_{t+1}, a_{t+1} \sim \pi(\cdot \mid q, \tau_t, T_t)$. Once T_t is fully populated, the complex query q can be answered by referring the evidence in T_t.

Unified View of InfoSeeking. This tabular formulation provides a unified representations of three distinct InfoSeeking paradigms: (1) Deep Search (Precise Filtering): The objective is to identify a unique candidate row that strictly satisfies all constraints ($|\mathcal{C}| > 0$), often involving complex multi-hop verification to filter out false positives; (2) Wide Search (Broad Aggregation): The primary goal is to gather required information ($|\mathcal{I}| > 0$) for a massive candidates, typically under minimal constraints (Wong et al., 2025); (3) DeepWide Search (Hybird): A complex hybrid scenario requiring the maximization of candidate discovery subject to strict constraint satisfaction, followed by dense information collection ($|\mathcal{C}| > 0, |\mathcal{I}| > 0$).

4 Implementation of TaS Framework

We instantiate the TaS framework as a multi-agent system centered around a shared, structured database table. As outlined in Algorithm 1 and Figure 8, the execution follows a three-phase process.

Table Initialization. The Planner parses the user query q and initialize the table structure in the database (`ConstructSchema`).

Dynamic Orchestration. In the main loop (Lines 4-18), the Planner Main-Agent dynamically selects the action: (1) **Row Expansion (Lines 6-10)**: For example, if the table lacks candidates, or if current candidates fail to satisfy query constraints, it formulates n diverse search strategies using the constraints (Line 7). These strategies are orchestrated to Sub-Agents in parallel to perform broad searches, aiming to discover new candidates; (2) **Cell Population (Lines 11-16)**: Conversely, if candidates are sufficient but their information is incomplete, the system transitions to this mode. Leveraging the independence of candidates,

Algorithm 1: Multi-Agent System of TaS

Input : Query Q, MaxSteps T_{max}, Timeout τ
Output : Final synthesized answer A
`// Phase 1: Table Initialization`
`// Define Key Cands./Cons./Info columns`
1 $S \leftarrow$ MainAgent.ConstructSchema(Q);
2 $Table \leftarrow$ Initialize(S); $State \leftarrow$ Pending
3 $\mathcal{H}_T \leftarrow \{\}$
`// Phase 2: Dynamic Orchestration`
4 **while** $State \neq$ *Done* $\wedge \neg$*Limits*(T_{max}, τ) **do**
5   $Plan \leftarrow$ Main.FormulateStrategy($Table, Q$)
6   **if** $Plan.action ==$ *ExpandRows* **then**
    `// Case: No enough valid candidates`
7     $\{q\}_{i=0}^{n} \leftarrow$ MakeQuery($Table$.ConsCols)
8   **foreach** q_i ***in parallel*** **do**
9       $Cands \leftarrow$ SubAgent.DeepSearch(q_i)
10       $Table$.AppendRows($Cands$)
11   **if** $Plan.action ==$ *PopulateCells* **then**
    `// Row-Level Parallel Execution`
12     $Rows \leftarrow Table$.GetIncompleteRows()
13   **foreach** $R_i \in Rows$ ***in parallel*** **do**
14       $q_i \leftarrow$ MakeQuery($R_i, Table$.EmptyInfoCols)
15       $Res_i \leftarrow$ SubAgent.DeepSearch(q_i)
16       $Table$.UpdateRow(R_i, Res_i)
17   $State \leftarrow$ Main.CheckSaturation($Table$)
18   Main.Update($\mathcal{H}_T$)
`// Phase 3: Answer Synthesis`
19 **return** Main.Synthesize($Table, Q$)

the Main Agent dispatches Sub-Agents in parallel to populate cells for each candidate. Notably, TaS allows for high flexibility: Since Sub-Agents inherently align with the recent specialized deep search models (Team et al., 2025b; Li et al., 2025a), TaS can seamlessly integrate advanced off-the-shelf search agents as sub-agents. Both Main-Agent and Sub-Agent manipulate table (`AppendRow` in Line 10 and `UpdateRow` in Line 16) via database interface. More details are in Appendix A.

Answer Synthesis. Upon detecting a saturated table state (or timeout), the Planner retrieves the structured evidence from the database to synthesize the final response A. For example, for Deep Search, the planner utilizes the filled table to cross-verify constraints for a precise conclusion; conversely, for Wide Search and DeepWide Search, it directly executes SQL queries to export the verified candidates.

5 Experimental Setup

5.1 Benchmarks and Metrics

To rigorously evaluate TaS across distinct long-horizon agentic infoseeking, we employ three categories of benchmarks: (1) **Deep Search**: We utilize GAIA (text-only) (Mialon et al., 2023) and BrowseComp-ZH (Zhou et al., 2025) to assess multi-step reasoning and precise filtering capabilities. Performance is measured by Accuracy, evaluated via standard LLM-as-a-Judge protocols (Zhou et al., 2025); (2) **Wide Search**: We employ the WideSearch benchmark to evaluate broad information aggregation (Wong et al., 2025). To reduce the randomness, we report the stable Avg@4 metrics of Column-F1 (Candidate Acc.), Row-F1 (Row-level Acc.), Item-F1 (Cell-level Acc) and Success Rate (SR, Table-level Acc.); (3) **Deep-Wide Search**: As existing benchmarks lack scenarios requiring both extensive candidate discovery and deep constratins verification and information collection, we curate a benchmark consisting of 20 challenging long-horizon InfoSeeking queries derived from real-world E-commerce scenarios (e.g., sourcing merchants meeting strict criteria). Given the high cost of expert curation, this dataset size aligns with concurrent studies (Parallel AI Team, 2025). Cases can be found in Figure A.1. We employ expert annotation to report Column-F1 and Item-Precision (Information Correctness) due to the open-ended complexity.

Experimental Scale and Cost. Some may argue for broader benchmark coverage. However, given the prohibitive cost of long-horizon execution (over \$5,000), our setup ensures a representative evaluation while maintaining computational feasibility.

5.2 Baseline Models and Systems

We compare TaS against two kinds of baselines: (1) **Agentic Frameworks**: We evaluate standard Single-Agent ReAct (ReAct-SA) (Yao et al., 2022; Tao et al., 2025), Multi-Agent ReAct (ReAct-MA) (Wong et al., 2025; Kim et al., 2025), and their compute-scaled variants (Zhu et al., 2025). Multi-Agent serves as the state-of-the-art baseline in Wide Search (Wong et al., 2025) and Deep Search (as evidenced in Table 1). These frameworks are instantiated with diverse foundation models, including GPT-5, Claude-Sonnet-4, Gemini-2.5 series, KIMI-K2 (Team et al., 2025a), Qwen3 series (Yang et al., 2025), etc.; (2) **State-of-the-Art Systems**: We further benchmark against specialized search agents, including commercial systems (Gemini DeepResearch) and models trained by Agentic RL (Team et al., 2025b; Tao et al., 2025).

5.3 Implementation Details

Our experiments are based on the SmolAgent framework (Roucher et al., 2025) and WideSearch (Wong et al., 2025). All agents utilize two standard tools: Google Search and Webpage Visit (Wong et al., 2025) to interact with environments. All training-based search sub-agents are served on a cluster of 8 NVIDIA A100 GPUs. To handle long contexts, we set the maximum context window to 64k tokens. We integrate webpage and context summarization strategies for reducing cost (Team et al., 2025b; Wu et al., 2025). Full hyperparameters, prompt details and table tool implementation in TaS are provided in Appendix A.

6 Main Results

This section provide experimental results on three kinds of Agentic InfoSeeking benchmarks: (1) Deep Search (Section 6.1); (2) Wide Search (Section 6.2) and (3) DeepWide Search (Section 6.3).

6.1 Results on Deep Search Benchmarks

Table 1 and Table 2 presents the comparative analysis on GAIA and BrowseComp-ZH benchmarks.

Model / System	Type	GAIA	BC-ZH
Foundation Models with Tools			
OpenAI Deep Research	-	67.4	42.9
GPT-5 High-Think	-	76.4	63.0
Claude-4-Sonnet (Thinking)	SA	68.3	29.1
Gemini-2.5-Pro	SA	60.2	27.8
Training-based Search Agents			
Tongyi DeepResearch (30B)	SA	70.9	46.7
MiroThinker-v1.0-8B	SA	66.4	40.2
MiroThinker-v1.0-30B	SA	73.5	47.8
MiroThinker-v1.0-72B	SA	81.9	55.6
Our proposed TaS Framework			
GPT-5 Medium-Think	SA	66.0	56.5
GPT-5 Medium-Think	MA	71.8	62.9
GPT-5 Medium-Think (Ours)	MA	**77.7**	**63.7**
Qwen3-Max	SA	39.8	23.5
Qwen3-Max	MA	**52.0**	34.3
Qwen3-Max (Ours)	MA	49.0	**35.3**
Gemini-2.5-Flash	SA	16.3	26.6
Gemini-2.5-Flash	MA	38.4	28.4
Gemini-2.5-Flash (Ours)	MA	**52.4**	**34.9**

Table 1: Performance Comparison on **Deep Search Benchmarks**. BC-ZH refers to BrowseComp-ZH.

TaS Outperforms Unstructured Baselines. TaS consistently outperforms most Single-Agent and Multi-Agent ReAct baselines across diverse backbone models. Most notably, when instantiated with the cost-efficient Gemini-2.5-Flash, our framework surpasses the Multi-Agent ReAct baseline by a substantial margin of +14.0% on GAIA (52.4% vs. 38.4%), outperforming better counterpart Qwen3-Max. This result confirms that the performance bottleneck in weaker models is often not reasoning capability, but search state management. By maintaining the search state into a structured table, TaS effectively enables smaller models to perform on par with significantly larger counterparts.

Superiority in InfoSeeking Setup. We observe a slight regression on GAIA (49.0% vs. 52.0%). However, the breakdown in Table 2 reveals that this drop is strictly confined to non-search tasks (-18.2%), where the structured table overhead is unnecessary for simple internal agentic tasks. Crucially, on the search-dependent subset central to our objective, TaS maintains its superiority (+2.5%).

6.2 Results on Wide Search Benchmark

Table 3 demonstrates the Avg@4 performance on WideSearch (Wong et al., 2025), which is suited to stress-test agents due to its massive search space (Avg. 274.8 table cells per query). Max@4 performance is shown in Table 9.

Superiority of TaS Framework. TaS demonstrates holistic superiority over state-of-the-art baselines. As shown in Table 3, TaS with `Claude-Sonnet-4 (NoThink)` achieves comparable performance to the ReAct-MA with `Claude-Sonnet-4 (Thinking)` on Success Rate (3.5% $\approx$ 3.6%). Besides, Max@4 Performance in Table 9 shows that TaS with `Claude-Sonnet-4 (NoThink)` significantly surpassing ReAct-MA (`Claude-Sonnet-4 (Thinking)`) on Success Rate (9.1% $>$ 6.5%), exhibiting higher potentional. Moreover, instantiated with the lightweight `Gemini-2.5-Flash`, TaS outperforms ReAct-MA baseline running on the much stronger `Gemini-2.5-Pro` (Success Rate: 2.2% $>$ 2.0%). This inversion indicates that in long-horizon tasks, the performance bottleneck shifts from reasoning capability to state management, where TaS's structured planning enables smaller models to rival sig-

Model	Sub-Task Type	Num	ReAct	Ours	Δ
Qwen3 -Max	Requires Search	80	46.8%	**49.4%**	+2.5%
	No Search	23	**68.2%**	50.0%	-18.2%
	Overall	103	**51.5%**	49.5%	-2.0%
Gemini 2.5-Flash	Requires Search	80	34.2%	**49.4%**	+15.2%
	No Search	23	55.0%	**60.0%**	+5.0%
	Overall	103	38.4%	**51.5%**	+13.1%

Table 2: Detailed Performance on GAIA. Please refer to Appendix D.2 for more details.

nificantly larger counterparts.

Model	ReAct Type	SR Acc	Row F1	Item F1	Col F1
Foundation Models with Tools					
Claude-S4 Think	SA	2.3	31.7	57.9	-
Claude-S4 Think	MA	3.6	38.5	62.2	-
Gemini-2.5-Pro	SA	1.5	30.0	51.0	-
Gemini-2.5-Pro	MA	2.0	33.5	57.4	-
OpenAI o3	SA	4.5	34.0	52.6	-
OpenAI o3	MA	5.1	37.8	57.3	-
KIMI-K2	SA	1.1	29.7	54.4	-
KIMI-K2	MA	3.0	36.2	61.2	-
WebLeaper	SA	4.0	31.0	48.8	-
Our proposed TaS Framework					
Gemini-2.5-Flash	SA	2.0	26.9	49.9	62.1
Gemini-2.5-Flash	MA	1.9	26.3	45.7	55.4
Gemini-2.5-Flash (Ours)	MA	**2.2**	**29.1**	**52.7**	**66.8**
Claude-S4 NoThink	SA	2.2	26.1	48.6	61.3
Claude-S4 NoThink	MA	3.2	33.7	56.6	68.0
Claude-S4 NoThink (Ours)	MA	**3.5**	**36.7**	**60.5**	**74.7**

Table 3: **Avg@4** Performance on **WideSearch benchmark**. Claude-S4 denotes `Claude-Sonnet-4`. Baseline results are copied from Wong et al. (2025), where Column-F1 scores are not recorded.

Better Precision-Recall Trade-off. Typically, expanding the search horizon in precision-recall trade-off, where aggressive exploration introduces noise and hallucinations. However, as shown in Table 4, TaS simultaneously improves both precision and recall performance. Specifically, TaS significantly boosts in Column-Recall (+8.4%) and Item-Recall (+6.9%) compared to the ReAct-MA. Crucially, this higher coverage does not come at the cost of precision (e.g. +4.4% in Item-Precision), validating the table constraints effectively filter out noise during the extensive information gathering.

Model	ReAct	Row	Item	Col
Precision Performance				
Claude-S4 NoThink	SA	31.0	54.6	75.5
Claude-S4 NoThink	MA	37.6	63.6	78.4
Claude-S4 NoThink (Ours)	MA	**39.6**	**68.0**	**84.6**
Recall Performance				
Claude-S4 NoThink	SA	23.6	44.6	56.0
Claude-S4 NoThink	MA	31.8	51.9	64.0
Claude-S4 NoThink (Ours)	MA	**34.2**	**58.8**	**72.4**

Table 4: Detailed Avg@4 Precision-Recall Performance of `Claude-Sonnet-4` on the WideSearch benchmark.

6.3 Results on DeepWide Search Benchmark

Models / Systems	ReAct	Col-F1	Item-P
Gemini DeepResearch	-	51.2	58.3
Claude-Sonnet-4	SA	39.5	35.2
Claude-Sonnet-4	MA	39.3	44.2
Our proposed TaS Framework			
Claude-Sonnet-4 (TaS)	MA	**55.9**	63.5
+ 32B Sub-Agent	MA	52.7	**67.7**

Table 5: Performance on **DeepWide Search Benchmark**. Baselines and TaS use `Claude-Sonnet-4`.

Superior Performance. On the challenging DeepWide benchmark, TaS demonstrates decisive superiority. As shown in Table 5, it outperforms not only ReAct-MA but also the state-of-the-art Gemini DeepResearch, achieving gains of +4.7% in Column-F1 and +5.1% in Item-Precision. This confirms that explicit structured planning provides a critical edge over proprietary black-box systems in complex long-horizon InfoSeeking tasks.

Flexibility and Efficiency. TaS further proves its architectural scalability by effectively decoupling planning from execution. As shown in the last row of Table 5, replacing the sub-agent with a fine-tuned 32B deep search model yields a promising result: while candidate discovery sees a marginal trade-off (Column-F1: 55.9% $>$ 52.7%), the information retrieval precision significantly improves (Item-Precision: 67.7% $>$ 63.5%). This result confirms that high-frequency search actions can be offloaded to cost-effective specialized model to boost precision, making TaS a highly flexible and efficient solution for industrial-scale applications.

7 Analysis

We investigate the underlying mechanisms of TaS through four critical research questions (RQs). Specifically, we examine whether structured planning enhances Robustness in long-horizon InfoSeeking (RQ1) and improves Efficiency beyond simple scaling search volume (RQ2). We further analyze the Test-Time Scaling (RQ3) and Ablation Studies (RQ4) to compare the planner versus the sub-agents. Detailed experimental setup and results are provided in Appendix A.4 and Appendix C.

7.1 Robustness on Long-Horizon InfoSeeking

RQ1: Is TaS robust to increasing complexity in long-horizon InfoSeeking? We classify instances in benchmarks into five difficulty levels based on distinct complexity metrics: constraint count $|\mathcal{C}|$ (searching complexity) for Deep Search, and table size (interaction horizon) for Wide Search. As visualized in Figure 2, TaS demonstrates widening superiority as complexity scales: (1) **Deep Search** (Top): The performance gap over baselines expands from +14.3% in Med-Hard to +17.9% in the Hard instances. Crucially, TaS maintains consistent accuracy levels that match or even exceed those of easier tiers, validating its stability in deep reasoning; (2) **Wide Search** (Bottom, `Claude-Sonnet-4`): The superiority of TaS is highlighted by the drastic expansion of the performance gap from Med-Hard (+1.7%) to the Hard tier (+13.3%). This divergence indicates that while baselines experience a complete breakdown ($>$ 30%), TaS exhibits a much slower rate of decay, effectively tracking search states.

7.2 Search and Exploration Efficiency

RQ2: Is performance driven by planning quality or strictly by search volume? To fairly test performance with comparable search efficiency, we categorize instances into five segments based on the number of tool usage (sorted by tool usage volume) and benchmark TaS against compute-scaled baselines: ReAct-MA with Majority Voting (MV, N=4) for Deep Search, and ReAct-MA (Max@4) for Wide Search. As shown in Figure 3, TaS demonstrates qualitative superiority over scaling variant of baselines: (1) **Deep Search**: For example, in the most demanding segment (Seg 5) of GAIA, TaS outperforms the ReAct-MA MV (+4.3% improvement) while strictly consuming fewer tool calls (Avg. 45.8 $<$ 53.5), proving that superior search

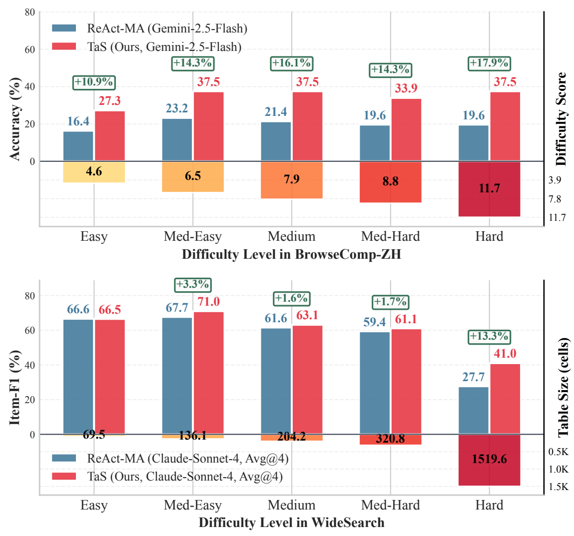

Figure 2: Robustness Analysis on BrowseComp-ZH (Top) and WideSearch (Bottom).

efficiency of TaS; (2) **Wide Search**: Similarly, TaS (Max@2) significantly outperforms ReAct-MA (Max@4) across all segments, while TaS' tool usage is comparable or even less. This confirms that TaS's advantage stems from precise and effective structured planning and state management, not merely increased search volume.

Moreover, TaS ensures precise exploration of the search space in WideSearch, as measured by Num@k (i.e., the maximum valid cells, defined as N_{total} $\times$ Item-P, achieved across k trials). Table 6 shows that TaS Num@1 already surpasses ReAct-MA Num@4 (199.7 $>$ 199.4). Besides, TaS Num@4 closely approaches the ground truth upper bound (251.1 vs. 274.8).

Method	Num@1	Num@2	Num@3	Num@4	GT
ReAct-SA	139.3	159.0	169.3	172.6	
ReAct-MA	158.0	186.0	194.7	199.4	274.8
TaS (Ours)	**199.7**	**211.4**	**229.4**	**251.1**	

Table 6: Comparison on Num@k of `Claude-Sonnet-4`. GT denotes the upper bound in ground-truth tables.

7.3 Test-time Scaling Analysis

RQ3: Does the structured planner drive more effective exploration during test-time scaling? We investigate whether allocating more inference compute benefits TaS more effectively than unstructured ReAct. Figure 4 illustrates the scaling trends on BrowseComp-ZH (Pass@N) and WideSearch (Max@N). It can be observed that as the compute budget (N) expands, the perfor-

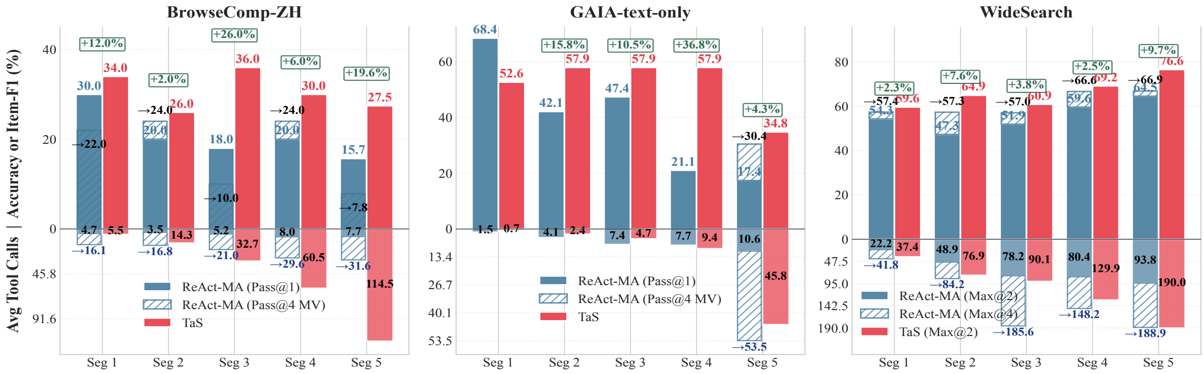

Figure 3: Search efficiency analysis of `Gemini-2.5-Flash` on Deep Search and Wide Search benchmarks.

mance gap widens. For instance, on BrowseComp-ZH, the performance gap between TaS and ReAct-MA widens from +2.4% (N=1) to +7.2% (N=2). On WideSearch, the advantage of TaS amplifies from +4.0% (N=3) to +4.4% (N=4). Besides, TaS at N=2 consistently exceeds ReAct-MA at N=3 (Deep Search) and N=4 (Wide Search). This demonstrates that TaS benefits more effectively from test-time scaling.

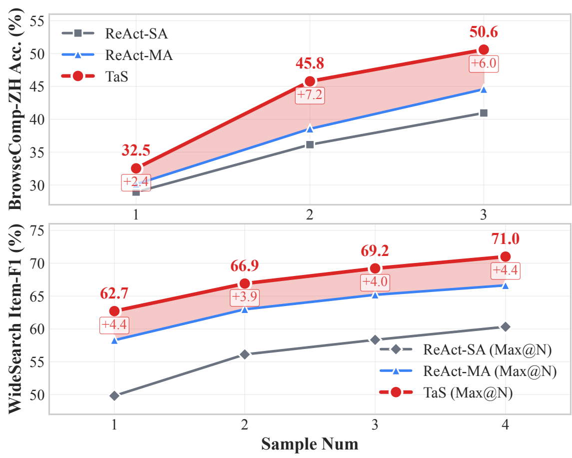

Figure 4: Test-time Scaling Analysis on BrowseComp-ZH (Top, `Gemini-2.5-Flash`) and WideSearch (Bottom, `Claude-Sonnet-4`).

7.4 Ablation Study on TaS Component

RQ4: Which component is the most critical: Planner Main-Agent or Sub-Agent? Table 7 reveals that the Planner Main-Agent is the critical bottleneck in our proposed framework: Downgrading the Planner from Qwen3-Max to Qwen3-30B-A3B causes a significant drop, while downgrading the Sub-Agent has a much milder impact.

Similar to findings in Section 6.3, TaS exhibits flexibility: As shown in the last four rows in Table 7, replacing the general `Gemini-2.5-Flash` Sub-Agent with the MiroThinker-8B deep search model (Team et al., 2025b) yields a substantial performance improvement across most metrics. This indicates that the Sub-Agent is "plug-and-play", allowing specialized and cost-efficient models to replace larger foundation models.

More importantly, integrating MiroThinker-8B into TaS (w/ `Gemini`) significantly outperforms the standalone `MiroThinker-8B` model on all metrics. This validates that TaS effectively unlocks and amplifies the potential of specialized deep search models, proving the effectiveness of planner in TaS.

Model Variant	DeepSearch	WideSearch		
	BC-ZH	Row-F1	Item-F1	Col-F1
TaS Framework: `Qwen3-235B-A22B` Sub-Agents. Planners are				
+ Qwen3-Max	**36.5**	**25.5**	**48.0**	58.1
+ Qwen3-235B	29.9	14.6	36.5	50.6
+ Qwen3-30B	7.1	8.6	22.9	33.3
Δ (Qwen3-30B)	**↓29.4%**	↓16.9%	**↓25.1%**	**↓24.8%**
TaS Framework: `Qwen3-Max` Planner. Sub-Agents are				
+ Qwen3-Max	**38.0**	**38.5**	**57.8**	**66.9**
+ Qwen3-235B	36.5	25.5	48.0	58.1
+ Qwen3-30B	27.0	16.9	45.0	63.6
Δ (Qwen3-30B)	↓11.0%	**↓21.6%**	↓12.8%	↓3.3%
TaS Framework: `Gemini-2.5-Flash` Planner. Sub-Agents are				
+ Gemini-2.5-Flash	33.0	**32.7**	52.5	65.8
+ MiroThinker-8B	**40.0**	32.1	**59.0**	**75.9**
Compare with `MiroThinker-v1.0-8B` Standalone Baseline				
Only MiroThinker	32.0	19.8	36.0	47.4
Δ (Only MiroThinker)	↓8.0%	↓12.3%	↓23.0%	↓28.5%

Table 7: Ablation study on the subsets of two benchmarks. The row (Δ) indicates the performance drop.

8 Conclusion

In this work, we introduced the Table-as-Search (TaS) framework that reformulates long-horizon agentic InfoSeeking as the Table Completion task.

TaS maps user query to structured table schema for precise tracking of search states. Extensive experiments demonstrate that TaS significantly outperforms state-of-the-art baselines across Deep, Wide, and DeepWide Search benchmarks. Furthermore, the framework exhibits superior robustness, efficiency, scalability and flexibility, paving the way for more robust InfoSeeking agents.

Limitation

Generalization to Non-Search Tasks. While TaS Framework excels in long-horizon InfoSeeking tasks, its applicability to general-purpose agentic tasks remains unstable. The structured tabular schema, optimized for external retrieval and state tracking, may introduce unnecessary rigidity for tasks relying solely on internal knowledge or simple instruction following. This limitation is evidenced by the performance fluctuations observed on non-search GAIA instances (Section 6.1), suggesting that future work should explore adaptive mechanisms to dynamically toggle between structured planning of TaS framework and flexible free-form reasoning based on task demands.

Relationship with Model Optimization. It is important to clarify that our contribution is architectural, orthogonal to recent advancements in model training or Agentic Reinforcement Learning (RL) (Li et al., 2025a; Team et al., 2025b; Tao et al., 2025). In this work, we do not perform specific fine-tuning for the TaS framework. However, our ablation studies (Section 6.2) reveal a promising synergy: existing training-based search agents (e.g., WebSailor (Li et al., 2025a), MiroThinker (Team et al., 2025b)) can be seamlessly integrated as Sub-Agents within TaS, boosting execution performance without architectural changes. This suggests that the Sub-Agents of TaS is plug-and-play compatible with the best open-source models. Consequently, the critical avenue for future work lies in optimizing the Planner Model. Developing specialized planners could further mitigate the dependency on proprietary models and fully unlock the potential of the TaS framework.

Dependency on Strong Planner. TaS's performance is currently bounded by the reasoning capability of the central Planner Main-Agent. As indicated by the ablation study (Section 7.1), while the execution layer (Sub-Agents) can be effectively offloaded to smaller, cost-efficient models without performance loss, the planning layer remains sensitive to model capacity. Downgrading the Planner to weaker models leads to significant performance degradation. Our future work will focus on optimizing the Planner—potentially through Agentic RL (Li et al., 2025a; Team et al., 2025b).

Distinction from Context Optimization. Our core contribution lies in structured planning to enhance search precision, rather than merely mitigating context overflow via compression. Consequently, recent context optimization strategies (e.g., summarization or folding) are **orthogonal** to our framework: TaS can also seamlessly incorporate them to further minimize token usage. However, distinct from these lossy compression methods, TaS offers a unique advantage by offloading critical search states to a structured external database. This inherently releases the agent's valuable context window for **complex reasoning** rather than passive information storage. Given this fundamental architectural distinction, comparing TaS against pure context compression baselines is unnecessary for validating the efficacy of structured planning.

Evaluation Scalability on DeepWide Search. A primary limitation of our curated DeepWide Search benchmark lies in the reliance on human evaluation. Unlike closed-domain tasks, DeepWide Search is inherently open-ended, rendering the construction of an exhaustive ground-truth universe computationally infeasible. To ensure manageable annotation costs, we explicitly constrain the retrieval target to a fixed quantity for each query (e.g., 30 candidates, as illustrated in Figure A.1). Consequently, accurate assessment currently necessitates human verification to validate whether retrieved candidates strictly satisfy complex constraints. To mitigate the prohibitive cost of annotation and improve efficiency, we implement a dynamic ground-truth maintenance strategy. Specifically, we construct a growing reference dataset by taking the union of verified correct matches (and maintaining an exclusion list for known false positives) across all evaluated systems and human annotation (Parallel AI Team, 2025). While this iteratively updates the ground truth to facilitate partial automation, the dependence on human-in-the-loop verification remains a constraint for large-scale reproducibility.

References

Guoxin Chen, Zile Qiao, Xuanzhong Chen, Donglei Yu, Haotian Xu, Wayne Xin Zhao, Ruihua Song, Wenbiao Yin, Huifeng Yin, Liwen Zhang, Kuan Li, Minpeng Liao, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025. Iterresearch: Rethinking long-horizon agents via markovian state reconstruction. *Preprint*, arXiv:2511.07327.

Tianqing Fang, Hongming Zhang, Zhisong Zhang, Kaixin Ma, Wenhao Yu, Haitao Mi, and Dong Yu. 2025. Webevolver: Enhancing web agent self-improvement with coevolving world model. *Preprint*, arXiv:2504.21024.

Google. 2025. Gemini deep research. `https://gemini.google/overview/deep-research/`. Accessed: 2025-12-28.

Kotaro Hara, Abi Adams, Kristy Milland, Saiph Savage, Chris Callison-Burch, and Jeffrey Bigham. 2017. A data-driven analysis of workers' earnings on amazon mechanical turk. *Preprint*, arXiv:1712.05796.

Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, and Weinan E. 2025. Pasa: An llm agent for comprehensive academic paper search. *Preprint*, arXiv:2501.10120.

Yubin Kim, Ken Gu, Chanwoo Park, Chunjong Park, Samuel Schmidgall, A. Ali Heydari, Yao Yan, Zhihan Zhang, Yuchen Zhuang, Mark Malhotra, Paul Pu Liang, Hae Won Park, Yuzhe Yang, Xuhai Xu, Yilun Du, Shwetak Patel, Tim Althoff, Daniel McDuff, and Xin Liu. 2025. Towards a science of scaling agent systems. *Preprint*, arXiv:2512.08296.

Tian Lan, Bin Zhu, Qianghuai Jia, Junyang Ren, Haijun Li, Longyue Wang, Zhao Xu, Weihua Luo, and Kaifu Zhang. 2025. Deepwidesearch: Benchmarking depth and width in agentic information seeking. *Preprint*, arXiv:2510.20168.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025a. Websailor: Navigating super-human reasoning for web agent. *Preprint*, arXiv:2507.02592.

Xiaoxi Li, Wenxiang Jiao, Jiarui Jin, Guanting Dong, Jiajie Jin, Yinuo Wang, Hao Wang, Yutao Zhu, Ji-Rong Wen, Yuan Lu, and Zhicheng Dou. 2025b. Deepagent: A general reasoning agent with scalable toolsets. *Preprint*, arXiv:2510.21618.

Tengxiao Liu, Zifeng Wang, Jin Miao, I Hsu, Jun Yan, Jiefeng Chen, Rujun Han, Fangyuan Xu, Yanfei Chen, Ke Jiang, and 1 others. 2025. Budget-aware tool-use enables effective agent scaling. *arXiv preprint arXiv:2511.17006*.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. *Preprint*, arXiv:2311.12983.

Parallel AI Team. 2025. Introducing findall api. `https://parallel.ai/blog/introducing-findall-api`. Accessed: 2025-12-26.

Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. 2024. Adapt: As-needed decomposition and planning with language models. *Preprint*, arXiv:2311.05772.

Aymeric Roucher, Albert Villanova del Moral, Thomas Wolf, Leandro von Werra, and Erik Kaunismäki. 2025. 'smolagents': a smol library to build great agentic systems. `https://github.com/huggingface/smolagents`.

Zhengwei Tao, Haiyang Shen, Baixuan Li, Wenbiao Yin, Jialong Wu, Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Liwen Zhang, and 1 others. 2025. Webleaper: Empowering efficiency and efficacy in webagent via enabling info-rich seeking. *arXiv preprint arXiv:2510.24697*.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, , and etc. 2025a. Kimi k2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

MiroMind Team, Song Bai, Lidong Bing, Carson Chen, Guanzheng Chen, Yuntao Chen, Zhe Chen, Ziyi Chen, Jifeng Dai, Xuan Dong, and 1 others. 2025b. Mirothinker: Pushing the performance boundaries of open-source research agents via model, context, and interactive scaling. *arXiv preprint arXiv:2511.11793*.

Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, and 1 others. 2025c. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*.

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

Ryan Wong, Jiawei Wang, Junjie Zhao, Li Chen, Yan Gao, Long Zhang, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, Wenhao Huang, Yang Wang, and Ke Wang. 2025. Widesearch: Benchmarking agentic broad info-seeking. *Preprint*, arXiv:2508.07999.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, and 1 others. 2025. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*.

An Yang, Anfeng Li, Baosong Yang, and etc. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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*.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, and 1 others. 2025. Agentfold: Long-horizon web agents with proactive context management. *arXiv preprint arXiv:2510.24699*.

Zhaoyang Yu, Jiayi Zhang, Huixue Su, Yufan Zhao, Yifan Wu, Mingyi Deng, Jinyu Xiang, Yizhang Lin, Lingxiao Tang, Yuyu Luo, Bang Liu, and Chenglin Wu. 2025. Recode: Unify plan and action for universal granularity control. *Preprint*, arXiv:2510.23564.

Guancheng Zeng, Xueyi Chen, Jiawang Hu, Shaohua Qi, Yaxuan Mao, Zhantao Wang, Yifan Nie, Shuang Li, Qiuyang Feng, Pengxu Qiu, Yujia Wang, Wenqiang Han, Linyan Huang, Gang Li, Jingjing Mo, and Haowen Hu. 2025. Routine: A structural planning framework for llm agent system in enterprise. *Preprint*, arXiv:2507.14447.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan O Arik. 2024. Chain of agents: Large language models collaborating on long-context tasks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhenyu Zhang, Tianyi Chen, Weiran Xu, Alex Pentland, and Jiaxin Pei. 2025. Recap: Recursive context-aware reasoning and planning for large language model agents. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, Yuxin Gu, Sixin Hong, Jing Ren, Jian Chen, Chao Liu, and Yining Hua. 2025. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *Preprint*, arXiv:2504.19314.

King Zhu, Hanhao Li, Siwei Wu, Tianshun Xing, Dehua Ma, Xiangru Tang, Minghao Liu, Jian Yang, Jiaheng Liu, Yuchen Eleanor Jiang, Changwang Zhang, Chenghua Lin, Jun Wang, Ge Zhang, and Wangchunshu Zhou. 2025. Scaling test-time compute for llm agents. *Preprint*, arXiv:2506.12928.

A Experimental Details

A.1 Benchmarks and Metrics

Deep Search Benchmark. We employ the standard LLM-as-a-Judge evaluation protocol from BrowseComp-ZH (Zhou et al., 2025) to assess the correctness of generated answers on both GAIA (Mialon et al., 2023) and BrowseComp-ZH benchmarks. For the ablation studies and efficiency analyses presented in Section 7, due to the high computational cost and API quota limitations, we utilize a representative subset of the BrowseComp-ZH dataset consisting of 100 randomly sampled instances.

Wide Search Benchmark. We adopt the official evaluation framework of the WideSearch benchmark (Wong et al., 2025) to reproduce the ReAct baselines and compute standard metrics, including Row-F1, Item-F1, and Success Rate. In addition to these metrics, we introduce a **Column-F1** metric to explicitly measure the accuracy of the retrieved entities within the table. This metric allows us to decouple the quality of entity discovery from the quality of information extraction. Similar to the Deep Search setting, experiments in Section 7 are conducted on a stratified subset of the WideSearch dataset containing 50 samples.

DeepWide Search Benchmark. The current research community lacks open benchmarks that simultaneously demand extensive horizontal breadth (identifying numerous entities) and vertical depth (complex constraints and attribute extraction). Such datasets are notoriously difficult to construct and evaluate. To address this gap, we follow Parallel AI Team (2025) to create a specialized DeepWide dataset. This dataset consists of 20 high-quality, complex samples focused on Business Development (BD) scenarios, which reflect real-world industrial workflows. As illustrated in Sample A.1, DeepWide Search presents significantly higher complexity than isolated Deep or Wide search tasks. They require a rigorous two-stage process: (1) Complex Filtering: The agent must scan massive amounts of information to identify entities that satisfy multiple strict constraints (e.g., target market, product category, pricing strategy); (2) Deep Information Collection: For each identified entity, the agent must perform deep searches to retrieve specific missing details (e.g., contact emails, executive names).

Given the inherently open-ended nature of these tasks, constructing an exhaustive ground-truth universe is computationally infeasible. To ensure robust yet manageable evaluation, we implemented a strict protocol: First, we explicitly constrain the retrieval target to a fixed quantity for each query (e.g., 30 candidates) to bound the search space. we construct the ground truth via a dynamic union strategy, aggregating verified correct matches from commercial state-of-the-art systems (like EXA.ai and Gemini DeepResearch etc.), our internal baselines (ReAct-MA and TaS), and expert annotation. To ensure reliability, the final ground truth was unified and verified by domain experts. This reference dataset is rigorously verified by domain experts, who also maintain an exclusion list for known false positives. This dynamic mechanism allows us to iteratively update the ground truth table, significantly reducing annotation costs while ensuring high-fidelity assessment for future evaluations.

Given the open-ended nature of these tasks, the experimental results are annotated by four experts engaged in business development (BD) applications, each holding at least a master's degree. The hourly wage of our human annotators is over $34, which is much higher than average hourly wage $3.13 on Amazon Mechanical Turk (Hara et al., 2017). We report two primary metrics for this benchmark: (1) **Column-F1:** Evaluates the accuracy of the identified entities against the complex constraints; (2) **Item-Precision (Item-P):** Measures the accuracy of the retrieved information specifically for the correctly identified entities.

Sample A.1: An Example of Our Curated Deep-Wide Search Benchmark

User Query: Please help me identify **30 merchants** that meet all the following criteria:

(1) Target the **Spanish market**;
(2) Sell **Adidas** sneakers;
(3) Offer **competitive pricing**;
(4) Possess mature **B2C operational experience**.

Required Information: For each identified merchant, retrieve the following contact details: *[Phone Number, Cooperation Email, Sales Platform, Official Website, CEO Name, Source URL].*

A.2 Fine-tuning Deep Search Sub-Agent

This section provides the details of our fine-tuned 32B model utilized in Section 6.3:

Base Model. We utilized Qwen3-32B as the backbone for our Deep Search Sub-Agent. This 32B-parameter scale offers the optimal trade-off between reasoning capability and computational efficiency compared to smaller (14B) or larger variants (72B).

Data Construction. We constructed a training dataset of approximately 12K samples using a hybrid strategy that combines trajectory distillation with reverse-synthesis to ensure diversity and robustness: (1) Trajectory Distillation (Forward): Following the trajectory collection paradigm of WebSailor (Li et al., 2025a), we collected multi-constraints user queries and distilled high-quality navigation trajectories. To ensure data quality, we implemented a rigorous iterative filtering pipeline. This involved removing unanswerable queries, employing a teacher LLM to parse and verify the format of search results, and optimizing the phrasing of questions based on ground-truth answers (hindsight relabeling). This yielded 11k high-quality samples; and (2) Reverse Synthesis (Reverse): To mitigate data sparsity for complex conditions, we employed a reverse-generation approach. We first sampled structured constraints to generate SQL queries and retrieve ground-truth candidates. These structured records were then converted into natural language templates and paraphrased into human-like complex search queries. This process contributed 1k samples specifically targeting multi-constraint reasoning.

Training Implementation. The model is trained using Supervised Fine-Tuning (SFT) within 64k context windows. Learning rate is 5×10^{-5} . The training is conducted on a computation cluster of 64 NVIDIA A100 GPUs within five hours.

Inference Settings. We set the a maximum context window of 32B model as 64K. This extended context capability is critical for maintaining global coherence during deep search sessions, allowing the agent to process extensive search results and retain long-term history without truncation.

A.3 Tools for Table Operation

Our tabular memory system is built on MongoDB with PyMongo interfaces to ensure scalable and persistent state management. We expose six atomic primitives for agent interaction:

- `create_table(schema)`: Initializes the table structure based on the query-derived schema.
- `add_records(data)`: Inserts new candidate entities (rows) discovered during the expansion phase.

- `update_records(filter, update)`: Modifies specific cells to populate missing attributes for targeted candidates.

- `show_table(limit)`: Serializes the current table snapshot into Markdown format for planner inspection.

- `count_table(filter)`: Returns the number of rows matching specific criteria to verify target quantity.

- `filter_records(query)`: Retrieves subsets of records (e.g., rows with empty cells) to isolate pending tasks.

All data manipulation operations (insertion, updates, and filtering) strictly adhere to standard PyMongo syntax (e.g., utilizing operators like $set, $exists). This enables the agent to perform precise logical queries natively within the database.

A.4 Experimental Setup for Analysis

Computing Complexity. To rigorously evaluate model performance across varying degrees of task difficulty, we classify the samples in Deep Search (BrowseComp-ZH) and Wide Search (WideSearch) benchmarks into five distinct difficulty categorizations: *Easy, Med-Easy, Medium, Med-Hard*, and *Hard*. The specific complexity metrics for each benchmark are defined as follows: (1) **Deep Search:** We quantify complexity based on the number of search constraints within the user query. We utilized `Gemini-2.5-Flash` to parse each query and enumerate these constraints. A higher constraint count necessitates more intricate multi-hop reasoning and stricter information filtering, thereby increasing task difficulty; (2) **Wide Search:** We determine difficulty based on the size of the ground-truth table (the number of the table celss). Larger tables inherently demand a higher volume of search interactions to achieve full coverage, directly corresponding to a longer interaction horizon.

Experiments on Subset. Due to limited API quotas, test-time scaling and ablation study are conducted on the sampled subsets of 100 BrowseComp-ZH and 40 WideSearch samples.

B Detailed Process of TaS

The detailed process of our proposed TaS are shown in Figure 8, aligning with the Algorithm 1.

C More Experimental Results

C.1 Full Results on GAIA

Table 8 provides the complete results of GPT-5, Qwen3-Max and Gemini-2.5-Flash on GAIA samples. It can be found that TaS consistently outperforms state-of-the-art baselines, while its performance is instable on tasks that do not require searching.

Model	Sub-Task Type	ReAct	Ours	Δ
GPT-5 Medium Think	Requires Search	66.25%	**71.25%**	+5.0%
	No Search	**91.30%**	86.96%	-4.34%
	Overall	71.84%	**77.67%**	+5.87%
Qwen3 -Max	Requires Search	46.84%	**49.37%**	+2.53%
	No Search	**68.18%**	50.00%	-18.18%
	Overall	**51.49%**	49.50%	-1.98%
Gemini 2.5-Flash	Requires Search	34.18%	**49.37%**	+15.19%
	No Search	55.00%	**60.00%**	+5.00%
	Overall	38.38%	**51.52%**	+13.13%

Table 8: Detailed Performance on GAIA: samples requiring search or not ($N_r = 80$ and $N_{nr} = 23$).

Model	ReAct Type	SR Acc	Row F1	Item F1	Col F1
Foundation Models with Tools					
Claude-S4 Think	SA	5.0	41.9	66.7	-
Claude-S4 Think	MA	6.5	52.2	73.1	-
Gemini-2.5-Pro	SA	5.0	41.4	63.6	-
Gemini-2.5-Pro	MA	6.5	44.6	66.3	-
OpenAI o3	SA	9.0	44.1	62.3	-
OpenAI o3	MA	9.5	50.5	68.9	-
KIMI-K2	SA	3.5	41.4	65.1	-
KIMI-K2	MA	6.5	49.6	70.7	-
Our proposed TaS Framework					
Gemini-2.5-Flash	SA	**5.0**	41.1	64.8	78.0
Gemini-2.5-Flash	MA	4.5	42.3	61.7	71.4
Gemini-2.5-Flash (Ours)	MA	**5.0**	**45.7**	**67.6**	**82.2**
Claude-S4 NoThink	SA	4.5	38.1	60.9	74.1
Claude-S4 NoThink	MA	4.0	46.8	66.9	78.2
Claude-S4 NoThink (Ours)	MA	**9.1**	**49.0**	**71.0**	**84.4**

Table 9: **Max@4** Performance on WideSearch benchmark. Claude-S4 refers to `Claude-Sonnet-4`. SR denotes Success Rate. Results of baselines are copied from the paper (Wong et al., 2025), where their Column-F1 scores are not recorded.

C.2 Max@4 Performance on WideSearch

Beyond the stable Avg@4 metrics, we also analyze the Max@4 performance to assess the upper bound of agent capabilities in massive information aggregation. As detailed in Table 9, TaS consistently unlocks superior potential compared to unstructured ReAct baselines. Most strikingly, TaS instantiated with the standard `Claude-Sonnet-4 (NoThink)` achieves a Success Rate of 9.1%, significantly surpassing the computationally heavier Multi-Agent ReAct equipped with `Claude-Sonnet-4 (Thinking)` (6.5%). This suggests that structured planning and state management is more critical than internal chain-of-thought reasoning for massive long-horizon search. Furthermore, this architectural advantage allows smaller models to punch above their weight. The lightweight `Gemini-2.5-Flash` with TaS outperforms the much stronger `Gemini-2.5-Pro (Multi-Agent ReAct)` across key metrics, achieving higher Row-F1 (45.7% vs. 44.6%) and Item-F1 (67.6% vs. 66.3%). This confirms that TaS effectively decouples performance from pure model scale, offering a cost-effective solution for industrial applications.

C.3 Search and Exploration Efficiency

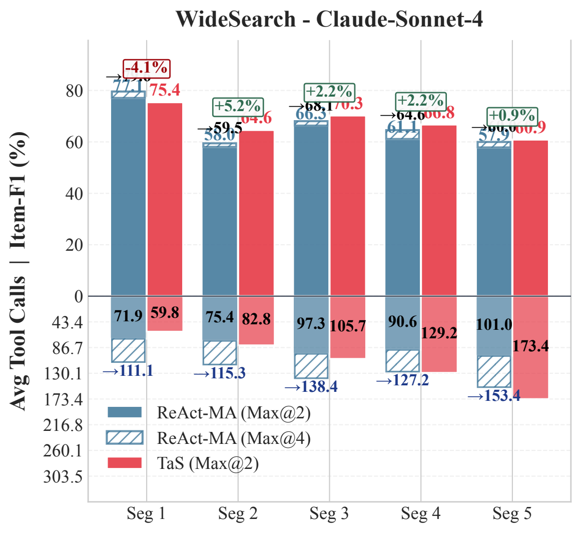

Figure 9: Search Efficiency Analysis on WideSearch of `Claude-Sonnet-4` model.

High performance in existing agents often comes at the cost of excessive interaction. However, Figure 9 reveals that TaS breaks this trade-off. On the WideSearch benchmark, TaS (`Claude-Sonnet-4 (NoThink)`) attains these performance gains with comparable or even lower tool usage volume than the Multi-Agent ReAct baseline. This demonstrates that the performance gains stem from structured planning precision rather than brute-force search scaling.

C.4 Robustness Analysis on WideSearch

Figure 10 demonstrates that TaS consistently outperforms the Multi-Agent ReAct baseline across all difficulty tiers. The advantage is most critical in the "Hard" setting, where the state space explodes to over 1,500 cells. While the baseline collapses to 21.4% Item-F1 under this cognitive load, TaS maintains robust performance at 32.3% (+10.9%). This confirms that structured planning effectively stabilizes small models against extreme context overload.

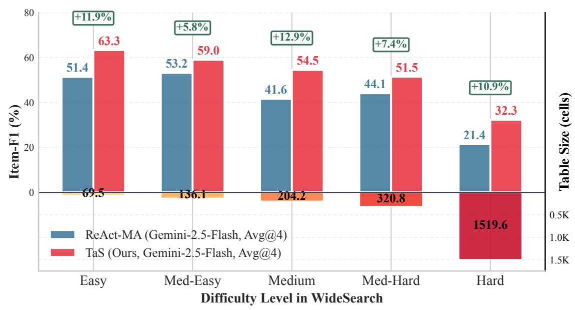

Figure 10: Search Efficiency Analysis on WideSearch of `Gemini-2.5-Flash` model.

D Case Study

D.1 Qualitative Analysis

Our case studies highlight how the table-centric design mitigates two critical failure modes of unstructured agents: (1) **Preventing Premature Convergence (Deep Search):** As shown in Figure 5, ReAct baselines often halt at partial matches (e.g., identifying "Hu Xia" but ignoring the album age). Our framework enforces **Global Verification** through schema filling, compelling the agent to validate every constraint against multiple candidates, thus filtering false positives; (2) **Eliminating Lazy Search (Wide Search):** As shown in Figure 6 and Figure 7, baselines struggle with long-horizon retrieval, resulting in missing rows and empty cells. In contrast, our planner ensures Completeness by decomposing the search space (e.g., by year) for row expansion and dispatching targeted sub-agents for cell population.

D.2 Search and No-Search Cases in GAIA

To evaluate our framework's adaptability, we stratified the GAIA validation set based on the ground-truth tool usage annotations provided in the dataset metadata. We identified 80 search-dependent samples (where the solution requires web interaction) and 23 no-search samples (where the solution relies solely on internal reasoning, calculation, or coding). Figure 11 contrasts the distinct behavioral requirements of these two categories.

Comparison on GAIA Samples

No-Search Task Query: *Given the operation * defined on set $S = \{a, b, c, d, e\}$ via the table below. Provide the subset of S involved in any counterexample proving * is not commutative. Format: alphabetical list.*

Search Task Query: *If all articles published by Nature in 2020 (articles only, not reviews) relied on statistical significance ($p = 0.04$), how many papers would be incorrect? Round up.*

Figure 11: Comparative Analysis on GAIA.

E The Use of Large Language Models

In preparing this manuscript, Qwen-Max and Gemini 3 are used solely as a writing assistant to improve grammar and clarity. The LLMs was not used for generating code, concepts, or any part of the core research methodology.

Case Study on BrowseComp-ZH #141

Query: *Find a singer who graduated from a university in their birth province. Around age 20, they sang a theme song for a TV drama with two identical characters in the title. They released their first album around age 22. Who is this singer?*

ReAct Baseline

Trajectory:

1. `Search`[Singer 20 years old TV theme song two identical characters...]
2. `Search`[Zhao Wei birth place graduation first album] $\rightarrow$ *Discarded*
3. `Search`[Chinese singer 20 years old theme song 22 years old first album...]
4. `Search`[Hu Xia TV theme song repeated characters]
5. `Visit`[.../item/Hu_Xia/...]
6. **Conclusion: ✗Hu Xia (Incorrect)**

Failure Analysis: The model found a partial match (Hu Xia) and halted prematurely. It failed to explicitly verify the "First Album Age" constraint (Hu released his first album at age 20, not 22), leading to a false positive.

Ours Proposed TaS Framework

Process Overview:

- **1. Schema Definition:** Columns defined for *Birth Prov., Univ. Prov., Theme Songs (Age $\approx$ 20), Album Year (Age $\approx$ 22).*
- **2. Search:** Retrieve 10 candidates (including *Jiang Dunhao, Chen Lin, Liu Xijun, Hu Xia, Shan Yichun...*).

3. Table Completion:

Candidate	Birth/Univ.	Match?	Theme Song (Age $\approx$ 20) & Album (Age $\approx$ 22)	Verdict
Hu Xia	GX / GX	✓	Song: *Summer Solstice* (Age 27) Album: *Hu Aixia* (2010, Age 20 $\neq$ 22)	✗
Shan Yichun	**ZJ / ZJ**	✓	**Song:** *Xu Xie* (Drama: *Yi Sheng Yi Shi*, 2021, Age 20) **Album:** *Brave Quota* (2022, Age 21*)	✓
Liu Xijun	GD / GD	✓	Song: *Bei Ke Feng Ling* (Age 18) Album: *Love Garden* (2010, Age 22) **Fails on song title constraint*	✗
...	...	...	...	...

**Note: Age 21 is considered "around age 22" by the ground truth standard.*

Correction Analysis: By explicitly filling the schema, the agent identified that only **✓Shan Yichun (Correct)** satisfied all constraints robustly, filtering out false positives like Hu Xia based on precise data points.

Figure 5: Case study between the ReAct and our proposed TaS Framework on the BrowseComp-ZH benchmark.

Case Study on WideSearch #EN-059

User Query: *Verify basic information for all TED Prize winners from 2005 to 2015. Required columns: [Year, Winner, TED Talk Title, Host City]. Output a Markdown table. Do not omit any cells; use "NA" if not found.*

Multi-Agent ReAct Baseline

Outcome (Low Recall):

Year	Winner	Talk Title	City
2005	Bono	NA	NA
2006	Lawrence Brilliant	...	Monterey
2007	Bill Clinton	NA	NA
...	...	...	...
2015	Dave Isay	*Everyone around ...*	Vancouver

Failure Analysis: Without a global schema to track progress, the agent **lost context** during the multi-hop reasoning. It inadvertently **omitted** the search for the critical "First Album Age" constraint, jumping directly to an erroneous conclusion based on incomplete evidence.

Our Proposed TaS Framework

Process Overview:

- **1. Schema & Strategy:** Schema defined as `[Year, Winner, Title, City]`. The planner explicitly decomposes the time range: *"Search 2005-2010 winners" and "Search 2011-2015 winners"*.
- **2. Row Expansion:** Parallel agents successfully retrieve all 11 winners (Rows) by cross-referencing multiple sources.
- **3. Cell Completion:** The planner detects missing "City" and "Title" values in the initial draft. Sub-agents are dispatched: e.g., `Search[Sylvia Earle TED Prize 2009 host city]`.

Final Outcome (100% Coverage):

Year	Winner	Talk Title	City
2005	Bono	*Three unusual ...*	Monterey ✓
...	...	...	...
2006	Cameron Sinclair	*A call for...*	Monterey ✓
...	...	...	...
2007	Bill Clinton	*Rebuilding Rwanda*	Monterey ✓
...	...	...	...
2009	Sylvia Earle	*Protect our oceans*	Long Beach ✓
...	...	...	...
2015	Dave Isay	*Everyone around you...*	Vancouver ✓

Conclusion: By structuring the search horizon and employing targeted cell-filling, our method achieves **11/11 recall** for rows and completes all attribute columns, whereas the baseline suffers from significant omission.

Figure 6: Case Study on WideSearch Benchmark (Task #EN-059).

Case Study on DeepWide Search (Task: US Lighting Merchants)

User Query: *Find 20 local US-based lighting manufacturers/merchants that operate on e-commerce platforms (Amazon, Walmart) or independent sites. Required columns: [Platform, Store Name, Email, Phone, Product Count].*

Multi-Agent ReAct Baseline

Outcome (Incomplete & Low Precision):

Merchant	Platform	Store Name	Email	Phone	Verdict
Progressive Lighting	Amazon/Site	Lights Online	NA	(866) 688-3562	✓
Brand Name Lighting	Amazon	*Generic Store*	NA	NA	✓
AvitaLights	Etsy	David Avital	NA	NA	✗*(Not US)*
HANM	Etsy	NA	NA	NA	✗*(Invalid)*
Wholesale Lighting	Amazon	NA	NA	NA	✗*(Not Found)*
...	...	...	...	...	...

Performance: Column-F1: 80.0% (False positives), **Item-P: 73.0%** (Missing contacts).

Failure Analysis: The baseline struggles with the **dual complexity** of breadth and depth. It fills slots with ineligible candidates (e.g., non-US Etsy sellers) to meet the "20 merchants" count and frequently fails to navigate to "Contact Us" pages for deep information extraction, resulting in empty email/phone cells.

Our Proposed TaS Framework

Process Overview:

- **1. Wide Search (Filtered Expansion):** Parallel sub-agents scan Amazon/Google, filtering out non-US sellers like *AvitaLights*.
- **2. Deep Search (Deep Crawling):** Targeted sub-agents visit official sites (e.g., *meyda.com*, *studio.hammerton.com*) to specifically locate contact details.

Final Outcome (High Recall & Precision):

Merchant	Platform	Store Name	Email	Phone	Verdict
Meyda Lighting	Site	Meyda.com	sales@meyda.com	800-222-4009	✓
LFI Lights	Amazon	Light Fixture Ind.	info@lightfixture...	877-534-4621	✓
Hammerton Studio	Site	Hammerton Studio	info@studio.ham...	801-973-8095	✓
HitLights	Amazon	HitLights	customer@hitli...	(855) 768-4135	✓
Commercial LED	Site	U.S. Wholesale	info@commercial...	(313) 528-7900	✓
LightArt	Site	LightArt	info@lightart.com	206-524-2223	✓
...	...	...	...	...	...
TorchStar	Site	TorchStar	info@torchstar.us	(800) 990-7688	✓

Performance: Column-F1: 95.0% (Correctly identified US merchants), **Item-P: 78.9%** (Rich contact details).

Conclusion: In DeepWide tasks, our framework excels by first strictly verifying candidate eligibility (US-based) during row expansion, and then leveraging deep search capabilities to retrieve hard-to-find attributes (Emails/Phones), significantly outperforming the baseline in both entity quality and information density.

Figure 7: Case Study in our curated DeepWide Search Benchmark.

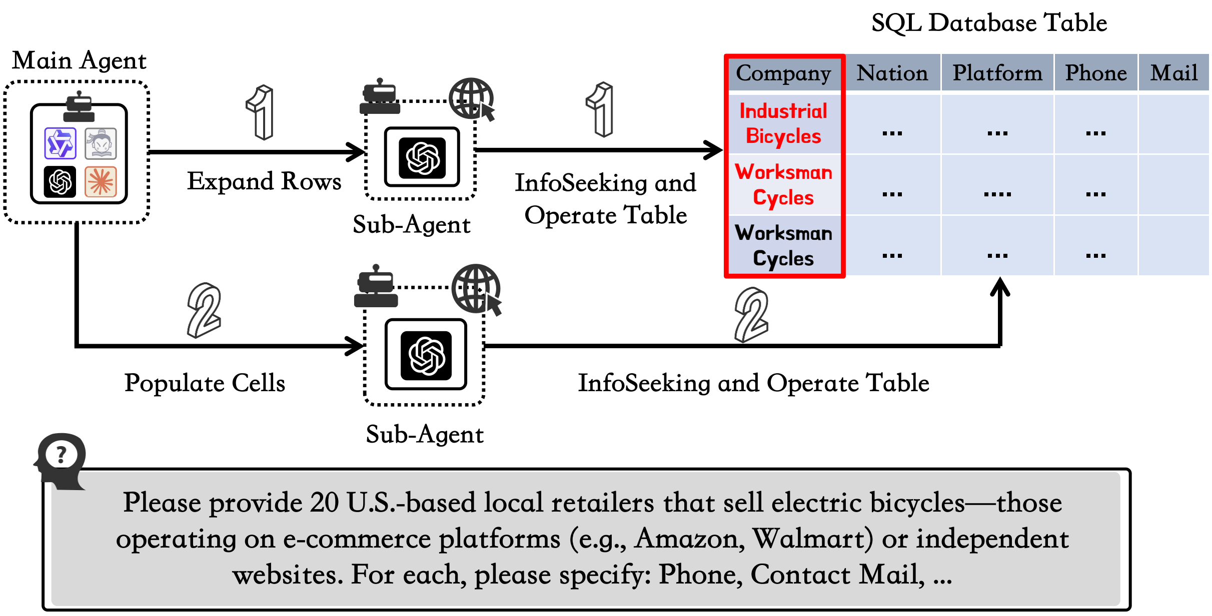

Figure 8: The detailed process of TaS on a complex DeepWide Search case in our benchmark.