

Lost in Distraction: LLM Planning Agents Recognize but Fail to Utilize Relevant Information under Context Noise

Weijie Liang^{1,†}, Yuanfeng Song^{2,†}, Xing Chen²

Caleb Chen Cao^{1,*}, Sirui Han^{1,*}, Yike Guo¹

¹The Hong Kong University of Science and Technology, Hong Kong, China

²ByteDance, China

Abstract

Large Language Model (LLM)-based planning agents operate under long, dynamically accumulated contexts that inevitably introduce noise. While it is commonly assumed that such noise impairs reasoning, our study shows that its primary impact lies elsewhere. Through a controlled noise injection framework on the DeepPlanning benchmark, we observe that noise does not significantly degrade the model’s ability to identify relevant information. Instead, it leads to systematic failures in utilizing that information, resulting in plans that are factually correct but violate specific user requirements. Noise also alters agent behavior, leading to reduced tool usage and fewer decision steps. To explain this discrepancy, we propose a two-stage framework for agent behavior under noise: information recognition and information utilization. Empirically, we identify distinct failure modes: some models fail to reliably identify relevant information, while others successfully identify it but fail to consistently apply it in decision-making. Furthermore, we find that selection mechanisms improve performance only when recognition is reliable, and can even be detrimental otherwise. These findings reveal a fundamental gap between recognizing and utilizing relevant information, highlighting consistent information utilization as a key bottleneck for developing robust LLM agents.

1 Introduction

In recent years, LLM-based agents have demonstrated strong capabilities in complex planning and decision-making tasks. By interleaving reasoning with external information retrieval, they achieve promising performance across tasks such as information seeking, code generation, and multi-step decision-making (Yao et al., 2022; Wei et al., 2025; Jimenez et al., 2024; Schick et al., 2023). This paradigm has further driven research on test-time scaling, where performance is improved by increasing reasoning steps or tool interactions.

However, this trend raises a fundamental question: *do longer reasoning trajectories necessarily lead to better decisions?* The execution of LLM agents is inherently an iterative process that interleaves information gathering and decision-making. During this process, a large number of intermediate states are continuously introduced into the context, including both task-relevant information and distracting content. As a result, the agent operates in a dynamically evolving noisy environment constructed by its own interaction

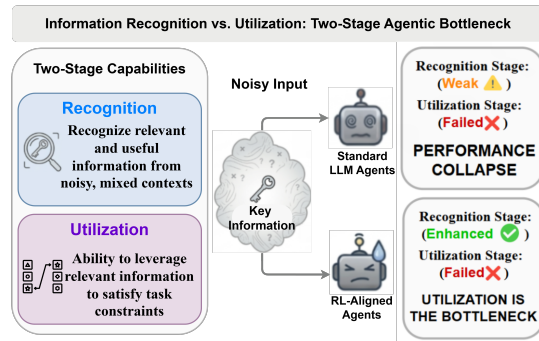


Figure 1: LLM agents process noisy context through two capabilities: recognition and utilization. While RL-Aligned agents improve recognition, *utilization* remains a key bottleneck.

[†]Equal contribution.

^{*}Corresponding authors.

history. Similar challenges arise in contextual compliance and legal-agent, where models must reason over imperfect contexts and consistently enforce scenario-specific requirements (Hu et al., 2025; Li et al., 2026a; Zhou et al., 2026). Prior work has shown that LLMs may struggle to effectively utilize information in long contexts, often overlooking relevant content due to position biases (Liu et al., 2024).

A natural intuition is that such noise would degrade the model’s reasoning ability. However, our findings suggest a more structured pattern. Specifically, we characterize the information processing of LLM agents under noisy environments as a two-stage process: **(1) information recognition, which identifies relevant information from mixed contexts, and (2) information utilization, which consistently applies this information during decision-making** (Fig. 1). Notably, LLMs with agentic alignment can significantly enhance information recognition, while information utilization remains a key bottleneck.

Based on a controlled framework built on the DeepPlanning benchmark (Zhang et al., 2026b), we find that even under high levels of semantically relevant noise, models maintain strong performance in the first stage, accurately identifying key facts with little degradation in reasoning ability. In contrast, the second stage exhibits clear instability: although models can retrieve correct information, they fail to consistently enforce task constraints during decision-making, resulting in plans that appear plausible but violate user requirements.

Furthermore, we observe that noise systematically weakens the agent’s exploration behavior (even the context length does not impose a significant burden), as reflected by simultaneous reductions in tool usage and reasoning steps. This suggests that agents tend to prematurely terminate the decision process rather than fail due to increased reasoning difficulty. Overall, the primary impact of noise does not lie in degrading reasoning ability, but in disrupting the stable transition from information recognition to decision execution.

To validate this two-stage perspective, we introduce a simple structural intervention: an explicit pre-decision filtering step that selects relevant information before execution. Experimental results show that this mechanism substantially restores performance under high-noise conditions. This further indicates that while models can recognize relevant information, they struggle to reliably utilize it in raw noisy contexts.

In summary, our work makes three main contributions: (1) we systematically study the effect of noise in dynamic contexts under a controllable noise injection framework based on DeepPlanning, and show that agents do not suffer a degradation in reasoning ability under noise, but instead tend to ignore task requirements; (2) we reveal a consistent shift in agent behavior under noise, where the overall policy remains largely unchanged but agents exhibit a tendency toward early termination; and (3) we propose a two-stage view of noise processing in LLM agents—recognition and utilization.

2 Related Work

Scaling Paradigms in Large Language Models. The scaling paradigms of LLMs can be categorized into three stages: training-time, test-time, and agentic scaling. Early work established relationships between model size, data, and compute. Specifically, *Scaling Laws for Neural Language Models* highlights the importance of balancing these factors (Kaplan et al., 2020), while subsequent research emphasizes optimizing compute allocation rather than increasing parameters (Hoffmann et al., 2022).

With the emergence of reasoning-oriented LLMs, research has shifted toward improving performance by allocating more computation during inference. Notably, *Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning* demonstrates that optimized test-time compute can outperform larger models (Snell et al., 2025). Further studies explore efficient exploration of these inference-time resources (Setlur et al., 2025).

Recently, scaling has extended to agent settings, where performance depends not only on internal reasoning but also on interaction steps and trajectories. Interaction can be treated as a distinct scaling dimension, where increasing the number of interaction steps improves performance (Shen et al., 2025). Budget-aware approaches further encourage agents to

take more steps by allocating tool-use budgets, enabling more extensive exploration under resource constraints (Liu et al., 2025). At the same time, recent work explores strategies such as parallel sampling, reflection, and verifier-based refinement (Zhu et al., 2025). However, emerging evidence also reveals fundamental limitations, including the context ceiling and the verification gap, suggesting that increasing interaction does not always lead to improved performance (Li et al., 2026b; Xu et al., 2026).

Benchmarks for Agentic Capability. In recent years, a variety of benchmarks have been proposed to evaluate LLM agents from different perspectives. One line focuses on tool usage and API interaction. For example, Tool Decathlon evaluates performance across diverse tool environments, while τ^2 -bench studies conversational agents under dual-control settings, and BFCL-V4 constructs structured function-calling scenarios for complex API compositions (Li et al., 2025b; Barres et al., 2025; Patil et al., 2025). These benchmarks are closer to real-world interfaces but typically emphasize local or single-step success. Another line considers closed interactive environments, such as vending machine simulations (Backlund & Petersson, 2025), which evaluate fully autonomous agent behavior over long horizons, requiring consistent policy execution and state tracking. A third category focuses on retrieval and open-environment search, including BrowseComp (Wei et al., 2025), BrowseComp-ZH (Zhou et al., 2025), and WideSearch (Wong et al., 2026). These tasks assess the ability to retrieve and integrate information from large-scale environments, but their dynamic and model-dependent nature makes it difficult to control noise and isolate its effects. Finally, planning benchmarks such as DeepPlanning target long-horizon decision-making under structured constraints (Zhang et al., 2026b). They require agents to satisfy both local and global constraints while making sequential decisions, jointly demanding long-term consistency and strong reasoning in a more controllable setting.

Agent Capability Enhancement. Beyond benchmark construction, recent work develops methods for improving agents’ context handling, interaction, and reliability. Memory-oriented frameworks compact long interaction histories to support long-horizon reasoning (Liang et al., 2026); contextual privacy work evaluates whether agents can satisfy context-dependent compliance requirements (Li et al., 2025a). In interactive visualization, logic-rule-guided multi-agent coordination, conversational generation, example selection, and prototype retrieval improve multi-step generation and refinement (Lu et al., 2026; Song et al., 2026; Li et al., 2025c). Recent deep-research work further co-evolves generation and evaluation under an adaptive evaluation environment (Zhu et al., 2026).

Although a large number of benchmarks evaluate LLM agents from various perspectives, and recent work explores agentic scaling strategies, systematic studies on how noise affects agent behavior remain limited; our work addresses this gap through a comprehensive evaluation using a controllable noise injection framework and a two-stage analysis of recognition and utilization.

3 Problem Formulation

While agent evaluation has shifted toward long-range tasks, existing benchmarks still face limitations in isolating the impact of noise on decision-making. **Tau2Bench** and **Tool Decathlon** primarily focus on tool-calling proficiency, while **BrowseComp** and **WideSearch** are often constrained by unstable real-time search environments. In contrast, **DeepPlanning** is specifically designed for long-range planning under complex constraints, featuring 240 test cases (equally divided between Chinese and English). It requires agents to proactively collect information from localized databases via tool calls while simultaneously managing local constraints (e.g., POI selection) and global boundaries (e.g., total budget). This controllable, localized environment makes it an ideal testbed for analyzing how injected noise impacts planning logic without the confounding variables of external search instability.

3.1 Formalizing Travel Planning as Constrained Search

We model the DeepPlanning task as a plan search problem. Given a candidate plan $\mathcal{P}$, the agent must construct a valid solution within the constrained environment.

Constraint Set. The global constraint set $\mathcal{C}$ is categorized into two types:

Commonsense constraints ($\mathcal{C}_{\text{com}}$): These evaluate whether a generated plan constitutes a reasonable, complete, and executable itinerary that adheres to general commonsense, thereby measuring plan reasoning ability. The constraints are organized into k semantic dimensions $\mathcal{D} = \{d_1, \dots, d_k\}$.

Personalized constraints ($\mathcal{C}_{\text{pers}}$): These evaluate whether the plan strictly satisfies the task-specific hard requirements (e.g., budget limits or designated hotels), thereby measuring constraint adherence capability. Thus, a plan can be locally coherent and executable under commonsense constraints while still violating global user-specific requirements.

Hierarchical Evaluation Function. We define the following scoring functions to evaluate the alignment quality of plan $\mathcal{P}$:

1. Commonsense Score (V_{com}). This measures dimension-level completeness. For each dimension d_i , the score is 1 only if all atomic constraints within that dimension are satisfied:

$$V_{\text{com}}(\mathcal{P}) = \sum_{i=1}^k w_i \cdot \left(\prod_{c \in d_i} \mathbf{1}(\mathcal{P} \text{ satisfies } c) \right) \quad (1)$$

2. Personalized Score (V_{pers}). We adopt a strict *all-or-nothing* criterion; any violation of personalized requirements results in a score of zero:

$$V_{\text{pers}}(\mathcal{P}) = \prod_{c \in \mathcal{C}_{\text{pers}}} \mathbf{1}(\mathcal{P} \text{ satisfies } c) \quad (2)$$

3. Commonsense Dimension Score ($V_{\text{com-dim}}$). To measure finer-grained sub-goal completion, we introduce $V_{\text{com-dim}}$. Unlike the binary truncation of V_{com} , it calculates the total proportion of satisfied atomic constraints across all commonsense dimensions, providing a more sensitive reflection of the model’s ability to maintain logical consistency under noise:

$$V_{\text{com-dim}}(\mathcal{P}) = \frac{1}{|\mathcal{C}_{\text{com}}|} \sum_{i=1}^k \sum_{c \in d_i} \mathbf{1}(\mathcal{P} \text{ satisfies } c) \quad (3)$$

3.2 Dynamic Noise Injection via Tool-based Sampling

To simulate the accumulation of noise in long-horizon decision-making, we modify the environment feedback at each step.

At time step t , when the agent executes an action a_t , the observation $\mathcal{O}_t$ is defined as:

$$\mathcal{O}_t = \text{Result}(a_t) \cup \mathcal{N}_t \quad (4)$$

where $\text{Result}(a_t)$ is the ground-truth response from a controlled local database, and $\mathcal{N}_t$ denotes the injected noise.

Noise Construction. The key requirement for $\mathcal{N}_t$ is that it should be **semantically plausible but decision-irrelevant**. To achieve this, we propose a isomorphic sampling strategy:

- **Isomorphic sampling:** Noise is generated by invoking the same tool as a_t (e.g., `SearchHotel`), but on data from other tasks.
- **Parameter shift:** The input parameters of the noise queries are sampled from a global pool, ensuring structural consistency with real outputs.
- **Semantic de-correlation:** We apply entity-level filtering to remove any overlap—where overlap refers to entities describing the same object—thereby avoiding explicit errors or misleading signals while preserving semantic similarity.

This design ensures that the injected noise closely resembles valid observations, while isolating its effect on **decision alignment** rather than factual correctness.

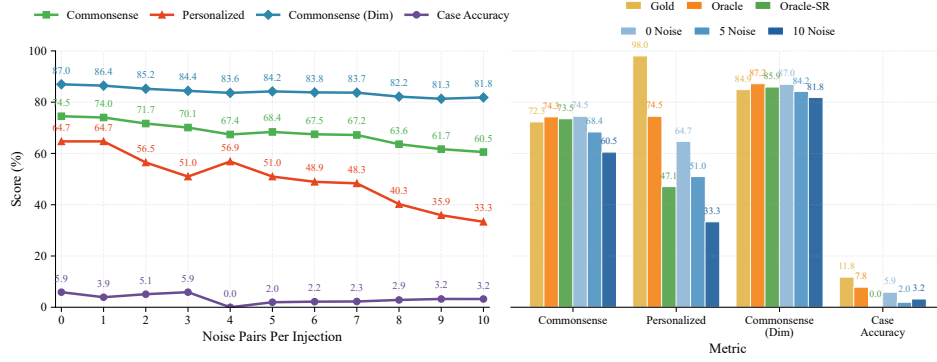


Figure 2: **Left:** Qwen3.5-397B performance under increasing noise levels on DeepPlanning. **Right:** Comparison of three additional settings (Gold, Oracle, Oracle-SR) providing a performance upper bound and isolating the recognition-utilization gap.

4 Effects of Noise on Agent Behavior

In this section, we analyze how contextual noise affects the behavior of LLM-based agents. Contrary to the common assumption that noise primarily degrades reasoning ability, our results reveal a more structured pattern: while reasoning remains relatively robust, constraint adherence deteriorates much more significantly.

We conduct experiments across a diverse set of frontier models, including GPT-OSS 120B, DeepSeek-V3.2, Kimi K2.5, Qwen3.5-397B, and Gemini 3 Flash Preview. In the main text, we focus on a representative comparison between Qwen3.5-397B and GPT-OSS 120B, distinguished by whether they have been further optimized via reinforcement learning for agentic behavior (Zhang et al., 2026a). The former represents models with agent-oriented optimization, while the latter serves as a baseline without such training. Other models exhibit similar trends to Qwen3.5-397B, and their results are provided in Appendix C.

4.1 Limited Impact of Noise on Reasoning

We show that noise does not significantly impair reasoning ability in the dynamic environment in agentic tasks. To support this, we examine its impact on information recognition and introduce three additional controlled settings to decouple inherent reasoning capacity from contextual interference:

- **Gold:** The agent is initialized with key information while still performing the full planning process, approximating an upper-bound performance.
- **Oracle:** the agent is provided with the full database with preserved structure, removing the need for autonomous exploration.
- **Oracle-SR (Shuffled Records):** all information is retained but randomly shuffled, creating a noise-like setting where the model must actively locate relevant content.

The detailed differences between *Oracle* and *Oracle-SR* are provided in Appendix B.

The left panel of Fig. 2 shows performance as the noise level increases. While the aggregated metric V_{com} exhibits a moderate decline, the fine-grained metric $V_{\text{com-dim}}$ remains highly stable. This discrepancy is mainly due to the *binary aggregation* of V_{com} , where a single violation nullifies the entire dimension score, rather than reflecting a true degradation in the model’s ability to satisfy individual constraints. The stability of $V_{\text{com-dim}}$ further suggests that noise has only a limited impact on individual constraint satisfaction.

Across the three additional settings, both V_{com} and $V_{\text{com-dim}}$ remain comparable to the 0-noise condition, suggesting that neither access to complete information nor structural shuffling fundamentally impairs the model’s reasoning ability. Instead, performance degradation arises primarily when noise is introduced into the agent’s iterative decision process.

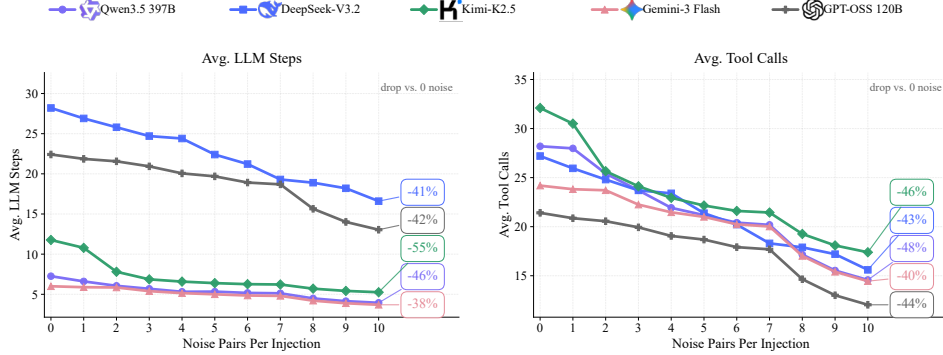


Figure 3: **Behavioral trends under increasing noise.** Average reasoning steps and tool calls decrease consistently across all models as noise increases.

Table 1: Comparison between semantically related noise and length/random-text controls. Each cell reports $V_{\text{pers}}/V_{\text{com-dim}}$ (%).

Model	0 Noise	10 Noise	Random text control	Length-padding control
Qwen3.5-397B	64.7 / 87.0	33.3 / 81.8	63.2 / 84.6	65.1 / 83.5
GPT-OSS-120B	17.6 / 51.6	8.0 / 48.0	18.1 / 52.4	15.2 / 53.2

Crucially, the agent can still satisfy the majority of atomic constraints even under noisy conditions, indicating that the primary bottleneck **does not lie in reasoning ability**, but in the **subsequent stage of decision control**, where necessary task constraints are often forgotten or inconsistently enforced.

4.2 Significant Degradation of Constraint Adherence under Noise

We observe a much more pronounced degradation in constraint adherence, despite relatively stable behavior on commonsense metrics.

In the **DeepPlanning** setting, we use the personalized score V_{pers} (Eq. 2) to evaluate whether generated plans satisfy the core task requirements. These requirements constitute the fundamental criteria of the task and are expressed as hard constraints (e.g., budget limits, mandatory conditions), making V_{pers} a direct measure of task consistency. As shown in Fig. 2, V_{pers} exhibits a clear **collapse** as noise increases, with a much sharper decline than the commonsense metrics (V_{com} and $V_{\text{com-dim}}$).

Even when plans are structurally reasonable, models often fail to satisfy key constraints. The three additional settings further reveal distinct failure patterns:

- **Gold:** V_{pers} is nearly saturated, as key constraint-relevant information is explicitly provided at initialization.
- **Oracle:** performance improves over the 0-noise setting, indicating that removing the need for exploration makes it easier for the model to satisfy constraints.
- **Oracle-SR:** despite having access to complete information (i.e., no missing data due to exploration), performance drops to the high-noise level, showing that unstructured context alone can significantly hinder constraint adherence.

Notably, in many failure cases (see Appendix E.1), generated plans remain factually plausible but violate core constraints. This phenomenon indicates that reasoning ability remains largely intact, while constraint adherence becomes unstable under noise.

To distinguish semantic distraction from context length effects, we compare 10 Noise with random text and length-padding controls under the same noise level. As shown in Table 1,

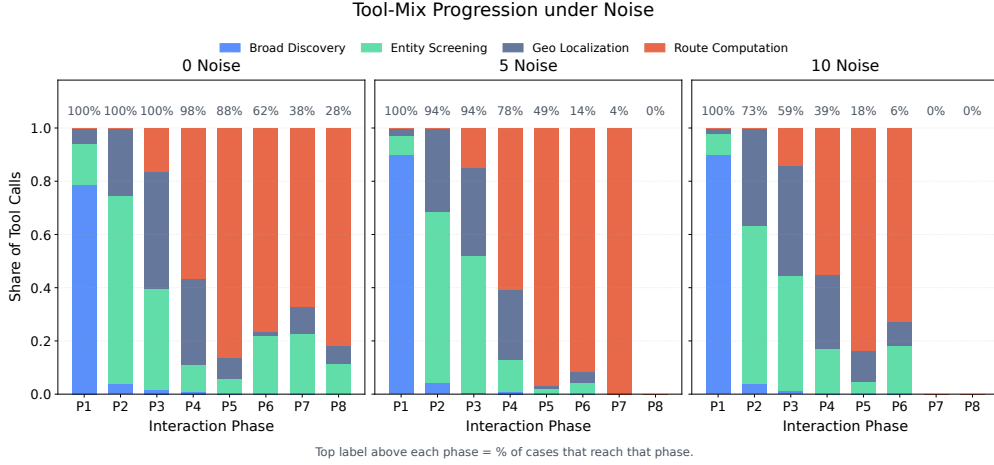


Figure 4: **Tool-mix progression across interaction phases under different noise levels.** Phases are defined by normalized trajectory progress; when the total number of steps is fewer than eight, each step is treated as a separate phase.

Table 2: **Context usage at the final step.** Absolute token counts and relative context window utilization across models.

Model	Context Limit	Average Tokens at Final Step (% Utilization)		
		0 Noise	5 Noise	10 Noise
Qwen 3.5 397B	256K	17,757 (6.9%)	68,265 (26.7%)	105,765 (41.3%)
DeepSeek-V3.2	160K	16,531 (10.3%)	67,123 (42.0%)	101,352 (63.3%)
Kimi-K2.5	256K	18,764 (7.3%)	68,100 (26.6%)	115,185 (45.0%)
Gemini 3 Flash	1M	17,525 (1.8%)	77,231 (7.7%)	124,261 (12.4%)
GPT-OSS 120B	128K	15,583 (12.2%)	68,738 (53.7%)	109,135 (85.3%)

these controls cause substantially smaller performance changes, indicating that degradation is mainly caused by semantically related distractions rather than longer contexts. This supports our design of schema-consistent and semantically plausible tool-observation noise, which challenges information recognition and utilization during agent planning.

4.3 Noise Induces Early Termination in Agent Behavior

Noise induces early termination in agent behavior, leading to reduced exploration despite stable planning policies. We analyze this effect by measuring reasoning steps and tool calls across varying noise levels (Fig. 3). As noise increases, both metrics consistently decrease across all models, indicating a tendency toward early termination.

To further characterize this behavior, we decompose each trajectory into normalized interaction phases and analyze the corresponding tool usage patterns (Fig. 4). We categorize tool calls into four functional groups: *Broad Discovery* (database exploration), *Entity Screening* (retrieving detailed information about specific entities), *Geo Localization* (grounding entities to spatial coordinates), and *Route Computation* (computing travel routes).

Across all noise levels, a consistent progression pattern can be observed: agents typically start with broad discovery, proceed to entity screening, then perform geo localization as a form of verification, and finally invoke route computation to construct the plan. This suggests that the overall planning policy remains stable under noise.

However, a key difference lies in the proportion of trajectories that successfully traverse this full pipeline. Under higher noise levels, significantly fewer cases reach later interaction

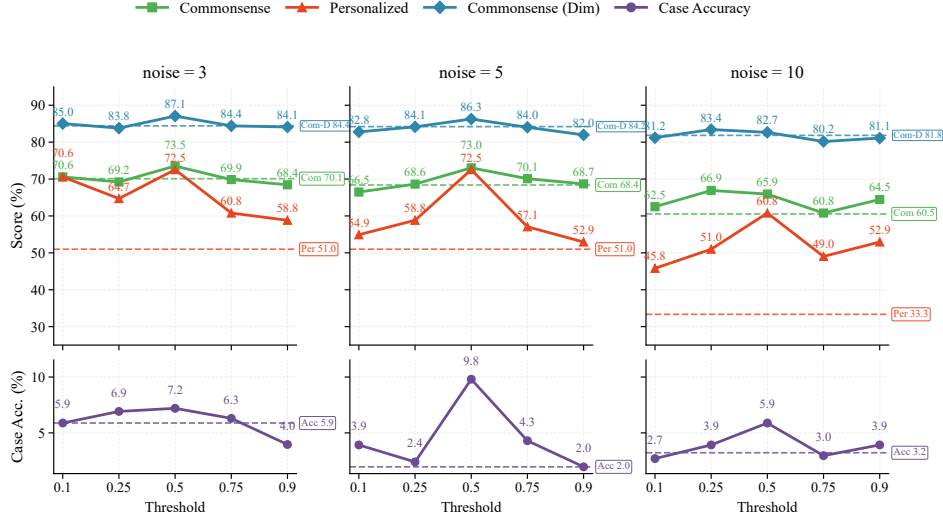


Figure 5: **Performance under selection thresholds (Qwen 3.5).** Results across different noise levels with varying selection thresholds. The x-axis denotes the threshold, and the y-axis shows the corresponding evaluation metrics.

phases, with most trajectories terminating at earlier stages. As a result, although the structural pattern of tool usage remains unchanged, the complete execution of the planning process becomes increasingly rare.

Finally, we examine the context resource usage at the final step (Tab. 2). Despite the increase in noise, the overall token consumption remains well within the context limits for most models, as reflected by the relatively low percentage of context window utilization. This indicates that the reduced interaction depth is not caused by context window saturation, but occurs even when sufficient context capacity is available.

4.4 A Two-Stage View: Recognition vs. Utilization under Noise

The preceding results reveal a consistent gap: while fine-grained information recognition (Eq. 3) remains relatively stable, overall task performance—especially constraint adherence (V_{pers})—drops sharply under noise. To explain this discrepancy, we decompose agent behavior into two capabilities: (1) **recognizing relevant information from noisy context**, and (2) **utilizing that information consistently during plan generation**.

To test this hypothesis, we introduce a lightweight **self-selection** step before final generation. Rather than compressing the noisy context into a summary—which may introduce hallucinations or discard useful details—we segment the accumulated interaction history into discrete items and ask the model to output only the indices of relevant ones. The selected items are then retained as the working context for subsequent planning, making self-selection a filtering step rather than an abstractive compression step.

We further use a dynamic threshold to control how often selection is triggered. Intuitively, a smaller threshold results in more frequent filtering, while a larger threshold delays it. The exact definition is provided in Appendix A. This intervention reveals two distinct patterns across models, supporting the separation between recognition and utilization.

Pattern 1: Utilization as the Bottleneck. For a substantial group of models, self-selection yields a strong performance recovery (Fig. 5). Under noise, these models can return close to their low-noise performance once the filtered context is provided. Since the relevant information is successfully preserved during selection, their recognition ability remains intact. The failure in the original noisy setting therefore lies primarily in utilization: the models identify the right information, but cannot reliably act on it when it is embedded in distracting context.

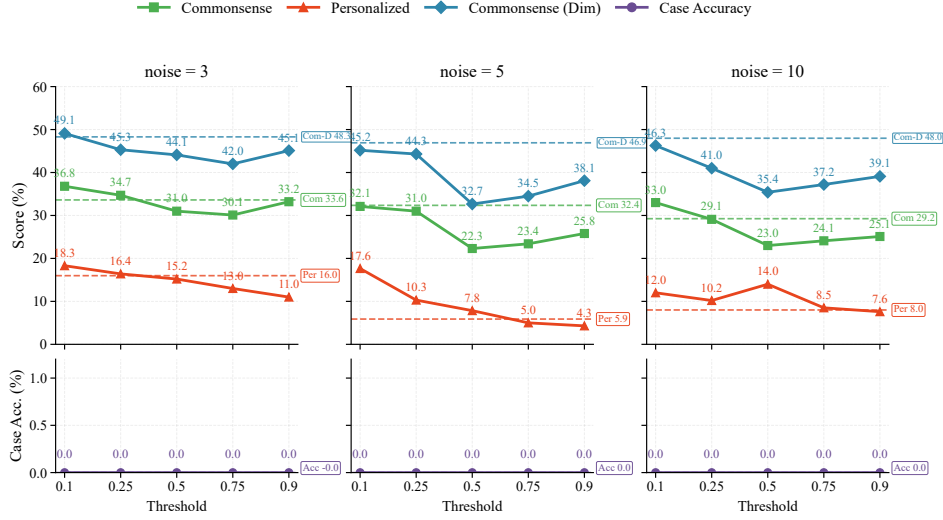


Figure 6: **Performance under selection thresholds (GPT-OSS-120B).** Results across different noise levels with varying selection thresholds.

Table 3: Comparison of self-selection with retrieval and compression baselines. “Noisy plan.” denotes direct planning over noisy observations. BM25, Emb., and Schema are rule-based filtering/compression methods, while “Summary” denotes LLM-based free-form summarization. Each cell reports $V_{\text{pers}}/V_{\text{com-dim}}$ (%).

Model	Noise	Noisy plan.	Selection	BM25	Emb.	Schema	Summary
Qwen3.5-397B	5	51.0 / 84.2	72.5 / 86.3	40.3 / 80.4	50.6 / 81.4	55.3 / 83.5	60.3 / 87.6
	10	33.3 / 81.8	60.8 / 82.7	50.8 / 84.5	48.6 / 84.4	40.8 / 81.4	60.5 / 83.6
GPT-OSS-120B	5	5.9 / 46.9	7.8 / 32.7	11.0 / 46.5	11.8 / 49.3	5.6 / 39.3	15.2 / 41.6
	10	8.0 / 48.0	14.0 / 35.4	12.4 / 39.3	10.5 / 41.9	15.0 / 44.2	5.0 / 25.6

Pattern 2: Instability in Both Stages. Other models show a different pattern, where selection does not produce stable gains (Fig. 6). Instead of recovering, V_{pers} remains volatile, while V_{com} and $V_{\text{com-dim}}$ further decline. This indicates that the bottleneck is not limited to utilization: recognition itself is unstable, and the additional reasoning overhead introduced by selection can further disrupt performance.

To further verify that the effectiveness of self-selection comes from model-based recognition rather than generic context reduction, we compare it with both retrieval-based methods and LLM-based summarization. Table 3 shows that rule-based retrieval and compression methods (BM25, Emb., and Schema) cannot consistently recover performance under noise. On Qwen3.5-397B, both LLM-based approaches (self-selection and summarization) substantially outperform non-generative filtering methods, suggesting that the model can recognize task-relevant evidence from noisy observations. In contrast, GPT-OSS-120B does not exhibit consistent gains from LLM-based filtering; in several cases, rule-based methods perform comparably or better. This indicates that when the model’s recognition ability is unstable, relying on its own evidence selection is not necessarily reliable.

Fig. 7 further illustrates this breakdown by plotting recognition against utilization for individual trajectories. The x-axis (Com-Dim) measures recognition quality, while the y-axis (hard-constraint pass ratio) measures utilization quality. If the two stages were tightly coupled, strong recognition would consistently translate into strong constraint satisfaction. Instead, we observe a clear decoupling under noise. Additional statistical tests in Appendix D further validate this observation: noise significantly degrades recognition quality, while self-selection provides significant recovery. In contrast, utilization remains comparatively stable under noise, although it is not completely unaffected.

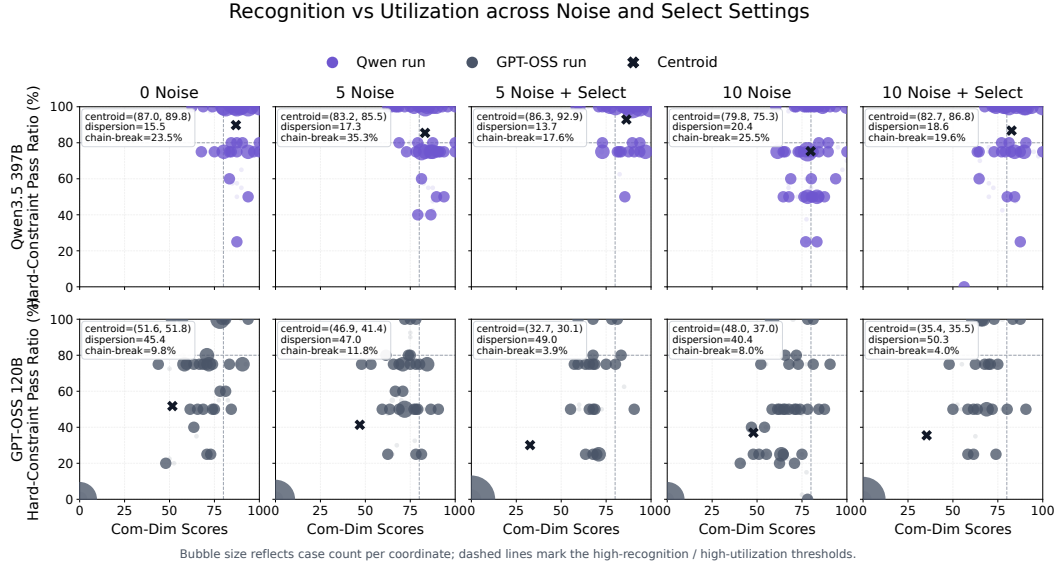


Figure 7: **Recognition vs. utilization across noise and selection settings.** Each point represents a trajectory, with the x-axis showing Com-Dim scores and the y-axis showing hard-constraint pass ratios. Bubble size indicates the number of cases sharing similar score coordinates.

For Qwen, the effect is particularly clear. As noise increases, the centroid shifts down and left, but the change is dominated by a drop along the utilization axis. The Com-Dim distribution remains comparatively stable, while hard-constraint performance shifts sharply downward and becomes more dispersed. This shows that the model largely retains recognition ability, but uses recognized information less reliably under noise. Self-selection substantially alleviates this effect, pulling the centroid back upward and reducing dispersion.

GPT-OSS shows a superficially similar shift under noise, but with a different failure mode. Both recognition and utilization are weaker and more dispersed from the outset, and self-selection does not reliably help—in some cases it even hurts performance. This suggests that the model suffers from instability in both stages, rather than a utilization-only bottleneck.

In summary, the significant disparity between these two types of models largely stems from whether they have undergone targeted agentic training or reinforcement learning. This suggests that specialized agentic training not only enhances the model’s task execution but also significantly contributes to the reliability of recognition in complex, noisy environments, thereby laying the foundation for consistent subsequent utilization.

5 Conclusion

This study shows that context noise in LLM planning agents does not primarily degrade reasoning ability, but instead disrupts consistent adherence to hard constraints during complex planning. By decomposing agent behavior into two stages—recognition and utilization—we find that models can often identify key constraints accurately, yet fail to reliably translate them into decision-making, resulting in plans that are commonsensical but violate user requirements.

Moreover, noise induces a behavioral shift toward early termination, with reduced tool use and shorter decision trajectories, indicating a breakdown in utilization stability. While a simple pre-decision filtering step can recover performance when recognition is reliable, it may harm models with unstable recognition. Overall, these findings highlight a fundamental gap between recognition and utilization, suggesting that improving robustness under noise requires addressing this gap, rather than solely enhancing basic reasoning.

Acknowledgments

This work is funded in part by the HKUST Start-up Fund (R9911), Theme-based Research Scheme grant (T45-205/21-N), the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government.

References

- Axel Backlund and Lukas Petersson. Vending-bench: A benchmark for long-term coherence of autonomous agents, 2025. URL <https://arxiv.org/abs/2502.15840>.
- Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2 -bench: Evaluating conversational agents in a dual-control environment, 2025. URL <https://arxiv.org/abs/2506.07982>.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. Training compute-optimal large language models. *ArXiv*, abs/2203.15556, 2022. URL <https://api.semanticscholar.org/CorpusID:247778764>.
- Wenbin Hu, Haoran Li, Huihao Jing, Qi Hu, Ziqian Zeng, Sirui Han, Xu Heli, Tianshu Chu, Peizhao Hu, and Yangqiu Song. Context Reasoner: Incentivizing reasoning capability for contextualized privacy and safety compliance via reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 865–883, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.44. URL <https://aclanthology.org/2025.emnlp-main.44/>.
- Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=VTF8yNQM66>.
- Jared Kaplan, Sam McCandlish, Thomas Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. Scaling laws for neural language models. *ArXiv*, abs/2001.08361, 2020. URL <https://api.semanticscholar.org/CorpusID:210861095>.
- Haoran Li, Wenbin Hu, Huihao Jing, Yulin Chen, Qi Hu, Sirui Han, Tianshu Chu, Peizhao Hu, and Yangqiu Song. PrivaCI-Bench: Evaluating privacy with contextual integrity and legal compliance. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10544–10559. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.acl-long.518. URL <https://aclanthology.org/2025.acl-long.518/>.
- Haoran Li, Yulin Chen, Huihao Jing, Wenbin Hu, Tsz Ho Li, Chanhon Lou, Hong Ting Tsang, Sirui Han, and Yangqiu Song. ContextLens: Modeling imperfect privacy and safety context for legal compliance. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6503–6518, San Diego, California, United States, 2026a. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.294. URL <https://aclanthology.org/2026.acl-long.294/>.
- Junlong Li, Wenshuo Zhao, Jian Zhao, Weihao Zeng, Haoze Wu, Xiaochen Wang, Rui Ge, Yuxuan Cao, Yuzhen Huang, Wei Liu, Junteng Liu, Zhaochen Su, Yiyang Guo, Fan Zhou, Lueyang Zhang, Juan Michelini, Xingyao Wang, Xiang Yue, Shuyan Zhou, Graham Neubig, and Junxian He. The tool decathlon: Benchmarking language agents for diverse, realistic, and long-horizon task execution. *CoRR*, abs/2510.25726, October 2025b. URL <https://doi.org/10.48550/arXiv.2510.25726>.

- Shuaimin Li, Xuanang Chen, Yuanfeng Song, Yunze Song, Chen Jason Zhang, Fei Hao, and Lei Chen. prompt4vis: prompting large language models with example mining for tabular data visualization. *The VLDB Journal*, 34, may 2025c. doi: 10.1007/s00778-025-00912-0.
- Xiaochuan Li, Ryan Ming, Pranav Setlur, Abhijay Paladugu, Andy Tang, Hao Kang, Shuai Shao, Rong Jin, and Chenyan Xiong. Benchmark test-time scaling of general llm agents, 2026b. URL <https://arxiv.org/abs/2602.18998>.
- Weijie Liang, Yuanfeng Song, Xing Chen, Caleb Chen Cao, Sirui Han, and Yike Guo. Vi-zoMem: A visual-textual memory framework for efficient long-horizon reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 7399–7422, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.365. URL <https://aclanthology.org/2026.findings-acl.365/>.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tac1.a_00638. URL <https://aclanthology.org/2024.tac1-1.9/>.
- Tengxiao Liu, Zifeng Wang, Jin Miao, I-Hung Hsu, Jun Yan, Jiefeng Chen, Rujun Han, Fangyuan Xu, Yanfei Chen, Ke Jiang, Samira Daruki, Yi Liang, William Yang Wang, Tomas Pfister, and Chen-Yu Lee. Budget-aware tool-use enables effective agent scaling. *ArXiv*, abs/2511.17006, 2025. URL <https://api.semanticscholar.org/CorpusID:283225884>.
- Jinwei Lu, Yuanfeng Song, Chen Zhang, and Raymond Chi-Wing Wong. Multivis-agent: A multi-agent framework with logic rules for reliable and comprehensive cross-modal data visualization. *Proc. ACM Manag. Data*, 4(1), feb 2026. doi: 10.1145/3786670.
- Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=2GmDdhBdDk>.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d842425e4bf79ba039352da0f658a906-Paper-Conference.pdf.
- Amrith Setlur, Matthew Y. R. Yang, Charlie Victor Snell, Jeremiah Greer, Ian Wu, Virginia Smith, Max Simchowitz, and Aviral Kumar. e3: Learning to explore enables extrapolation of test-time compute for LLMs. In *Second Workshop on Test-Time Adaptation: Putting Updates to the Test! at ICML 2025*, 2025. URL <https://openreview.net/forum?id=eIoACmXLN4>.
- Junhong Shen, Hao Bai, Lunjun Zhang, Yifei Zhou, Amrith Setlur, Shengbang Tong, Diego Caples, Nan Jiang, Tong Zhang, Ameet Talwalkar, and Aviral Kumar. Thinking vs. doing: Agents that reason by scaling test-time interaction. In *First Workshop on Multi-Turn Interactions in Large Language Models*, 2025. URL <https://openreview.net/forum?id=LkBUwX0y7b>.
- Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=4FWAwZtd2n>.
- Yuanfeng Song, Jinwei Lu, and Raymond Chi-Wing Wong. Covis: Neural and llm-driven multi-turn interactions for conversational text-to-visualization generation. *The VLDB Journal*, 35, nov 2026. doi: 10.1007/s00778-025-00954-4.

- Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents, 2025. URL <https://arxiv.org/abs/2504.12516>.
- Ryan Wong, Jiawei Wang, Junjie Zhao, Li Chen, Yan Gao, Zhanglong, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, Wenhao Huang, Yang Wang, and WANG KE. Widesearch: Benchmarking agentic broad info-seeking. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=Q7YUY7zGkZ>.
- Zelai Xu, Zhexuan Xu, Ruize Zhang, Chunyang Zhu, Shi Yu, Weilin Liu, Quanlu Zhang, Wenbo Ding, Chao Yu, and Yu Wang. Widesearch-r1: Exploring width scaling for broad information seeking via multi-agent reinforcement learning. *ArXiv*, abs/2602.04634, 2026. URL <https://api.semanticscholar.org/CorpusID:285285921>.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *ArXiv*, abs/2210.03629, 2022. URL <https://api.semanticscholar.org/CorpusID:252762395>.
- Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhong-Zhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita Velez, Yue Liao, Hongru WANG, Mengyue Yang, Heng Ji, Jun Wang, Shuicheng YAN, Philip Torr, and LEI BAI. The landscape of agentic reinforcement learning for LLMs: A survey. *Transactions on Machine Learning Research*, 2026a. ISSN 2835-8856. URL <https://openreview.net/forum?id=RY19y2RI10>. Survey Certification.
- Yinger Zhang, Shutong Jiang, Renhao Li, Jianhong Tu, Yang Su, Lianghao Deng, Xudong Guo, Chenxu Lv, and Junyang Lin. Deepplanning: Benchmarking long-horizon agentic planning with verifiable constraints, 2026b. URL <https://arxiv.org/abs/2601.18137>.
- 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. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese, 2025. URL <https://arxiv.org/abs/2504.19314>.
- Yujin Zhou, Yidan Huang, Sirui Han, and Yike Guo. How well can large language model agents simulate complex legal dispute resolution? *Artificial Intelligence and Law*, 2026. doi: 10.1007/s10506-026-09518-5. URL <https://doi.org/10.1007/s10506-026-09518-5>. Advance online publication.
- Han Zhu, Chengkun Cai, Yuanfeng Song, Xing Chen, Sirui Han, and Yike Guo. Self-Evolving Deep Research via Joint Generation and Evaluation. *arXiv preprint arXiv:2606.04507*, 2026. URL <https://arxiv.org/abs/2606.04507>.
- 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. Scaling test-time compute for llm agents. *CoRR*, abs/2506.12928, June 2025. URL <https://doi.org/10.48550/arXiv.2506.12928>.

A Self-Selection Setup

We implement a *self-selection* mechanism to control context growth under the noisy planning setting. Instead of summarizing history into free-form text, the system performs **entry-level selection**, retaining only a subset of interaction units as structured evidence. The dialogue history is flattened into fine-grained entries $\{e_1, \dots, e_n\}$ (e.g., tool calls, individual tool outputs, JSON objects, and intermediate reasoning). A separate compression model takes the original query and all entries, and outputs a subset $K \subseteq \{1, \dots, n\}$. The selected entries are concatenated into a ContextMemory block that replaces earlier history, while the most recent interaction segment is always preserved.

Selection is triggered by a threshold defined relative to a reference context scale from the noise experiments. For each case c and model m , we record the peak context length in the full-context (no-selection) trajectory:

$$P_m(c).$$

Given a ratio $\theta \in (0, 1]$, the trigger threshold is

$$T_m(c, \theta) = \lfloor \theta \cdot P_m(c) \rfloor.$$

Let $N_m(c, s)$ denote the current context length at step s ; selection is applied when

$$N_m(c, s) \geq T_m(c, \theta).$$

This formulation ties the trigger to the context scale observed in noisy full-context runs. The parameter θ thus controls when selection occurs relative to this reference, effectively adjusting the selection frequency across tasks and models.

Selection Prompt Below is the prompt used for selection:

Listing 1: Prompt for *Selection*.

```
You are a dialogue history selector for a travel planning agent.

Your task is NOT to rewrite or summarize the content.
Instead, you must select the interaction entries that should be retained
for future planning.

Rules:
- Prioritize entries that are directly relevant to the current task,
  including:
  facts, candidates, hard constraints, budget, time, locations,
  transportation, hotels, restaurants, and attractions
- Preserve intermediate conclusions and reasoning
- Remove irrelevant or noisy entries
- If uncertain about relevance, prefer keeping the entry
- If a tool result is kept, the corresponding tool call or decision entry
  should
  usually also be retained
- Do not fabricate, add, or alter any factual content

Output requirements:
- Return ONLY JSON in the format: { keep_ids : [ E1 , E2 ] }
- Only use entry IDs that appear in the input
- Do not include any additional explanation

User Prompt Template:
## User Task
{user_query}
## Full Interaction Entries
{entries_text}

Return only JSON: { keep_ids : [...] }
```

B Oracle and Oracle-SR Experimental Settings

The full context is constructed from multiple structured data sources, including `trains.csv`, `flights.csv`, `hotels.csv`, `attractions.csv`, `restaurants.csv`, and `locations_coords.csv`. Each file contains dozens to hundreds of entries with rich attribute fields (e.g., time, price, rating, and location).

For clarity, Fig. 8 and Fig. 9 present simplified visualizations that show only a subset of entries with lightly normalized textual formatting, while the underlying data is more structured and schema-consistent. In practice, the LLM is exposed to the full context in a more complete and structured JSON-like format, rather than the simplified representation shown here.

In the **Oracle** setting (Fig. 8), entries are grouped by source, forming contiguous blocks that preserve local coherence within each data type. In contrast, in the **Oracle-SR** setting (Fig. 9), entries are randomly interleaved across sources into a single sequence, breaking this structural locality while keeping the underlying information unchanged.

Oracle (Grouped by Source)

[System Prompt]
[Task Prompt]

trains.csv
[row0]: Hefei → Nanjing, Train T235, dep 02:32, arr 06:16, price ¥41.5
[row1]: Hefei South → Nanjing South, Train D3074, dep 15:09, arr 16:12, price ¥61
⋮

flights.csv
[row0]: Flight MU9085, Hefei Xinqiao → Fuzhou Changle, dep 08:20, price ¥979
[row1]: Flight MU2708, Fuzhou → Nanjing Lukou, dep 11:25, includes baggage
⋮

hotels.csv
[row0]: Home Inn Neo, near Nanjing Railway Station, 2-star, price ¥222/night
[row1]: Home Inn Business, Xinjiekou district, breakfast included, price ¥235/night
⋮

attractions.csv
[row0]: Nanjing Museum, historical-cultural site, rating 4.8/5, open 09:00–17:00, ticket free
[row1]: Deji Plaza, shopping/leisure complex, rating 4.8/5, open 10:00–22:00
⋮

restaurants.csv
[row0]: Yichun Restaurant, near museum, rating 3.9/5, avg cost ¥53, Jiangsu cuisine
[row1]: Taoxiangju Restaurant, rating 4.1/5, avg cost ¥124, local specialties
⋮

locations_coords.csv
[row0]: Zhongshan Mausoleum, coordinates (32.06N, 118.85E), landmark attraction
[row1]: Nanjing Museum, coordinates (32.04N, 118.82E)
⋮

Figure 8: **Oracle context layout.** All entries are grouped by source (e.g., trains, flights, hotels), preserving structural locality within each data type.

Oracle-SR (Shuffled Across Sources)

[System Prompt]

[Task Prompt]

[locations.coords.csv][row0]: Zhongshan Mausoleum, coordinates (32.06N, 118.85E)
 [restaurants.csv][row1]: Taoxiangju Restaurant, rating 4.1/5, avg cost ¥124
 [trains.csv][row0]: Hefei → Nanjing, Train T235, dep 02:32, arr 06:16, ¥41.5
 [flights.csv][row1]: MU2708, Fuzhou → Nanjing, dep 11:25, arr 12:50
 [hotels.csv][row0]: Home Inn Neo, 2-star, rating 4.7, ¥222/night
 [attractions.csv][row1]: Deji Plaza, rating 4.8, open 10:00–22:00
 [trains.csv][row1]: Hefei South → Nanjing South, D3074, 15:09–16:12, ¥61
 [flights.csv][row0]: MU9085, Hefei → Fuzhou, 08:20–09:50, ¥979
 [hotels.csv][row1]: Home Inn Business, 3-star, rating 4.8, ¥235/night
 [attractions.csv][row0]: Nanjing Museum, rating 4.8, 09:00–17:00
 [locations.coords.csv][row1]: Nanjing Museum, coordinates (32.04N, 118.82E)
 [restaurants.csv][row0]: Yichun Restaurant, rating 3.9, ¥53
 :

Figure 9: **Oracle-SR context layout.** The same entries are randomly interleaved across sources, breaking structural locality and introducing noise-like organization.

C Results for Remaining Models

DeepSeek-V3.2

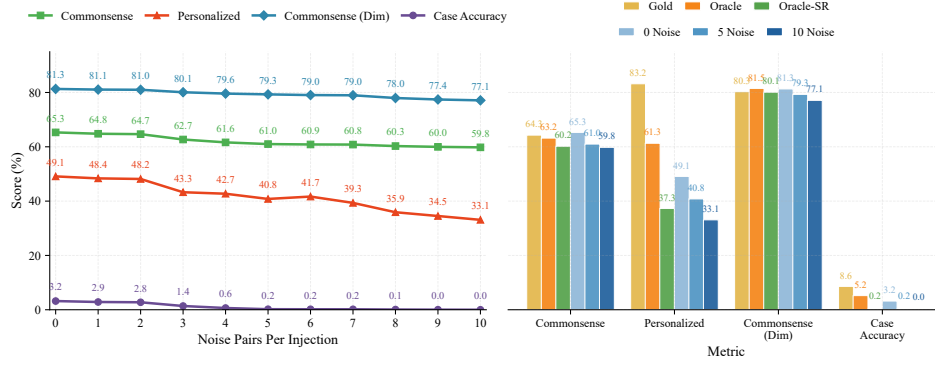


Figure 10: F DeepSeek-V3.2 under noisy planning. Left: Performance under increasing noise levels on DeepPlanning. Right: Comparison of the Gold, Oracle, and Oracle-SR settings.

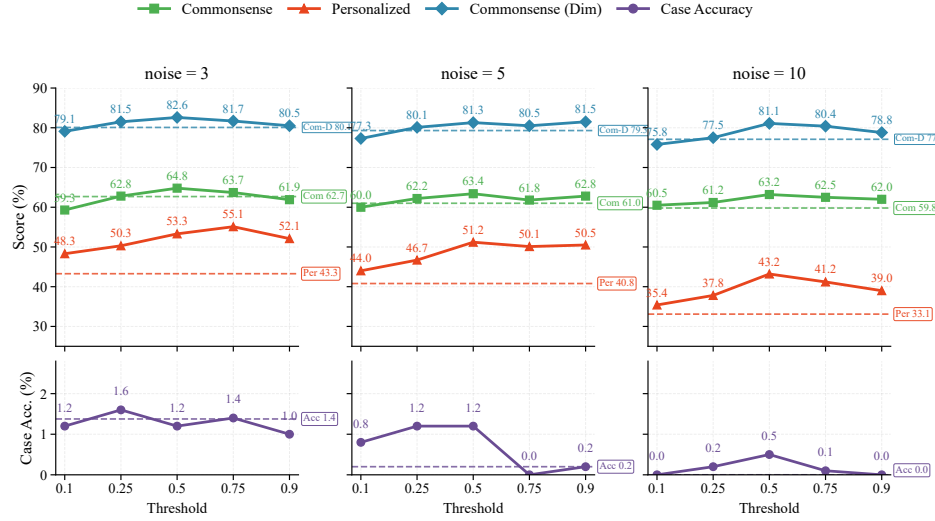


Figure 11: DeepSeek-V3.2 under different selection thresholds. Results across noise levels with varying self-selection trigger thresholds.

The above figures present the results of DeepSeek-V3.2. This model is trained with agentic alignment, and exhibits behavior consistent with Qwen3.5 in the main text.

As shown in Fig. 10, although the overall performance is slightly lower than Qwen3.5, the trends under both noise injection and additional settings (*Gold*, *Oracle*, and *Oracle-SR*) remain consistent. In particular, the degradation of V_{pers} under noise and its recovery in controlled settings follow a similar pattern. Notably, V_{com} appears more stable across noise levels compared to Qwen3.5.

Fig. 11 further shows that, consistent with Qwen3.5, introducing self-selection leads to a substantial improvement in V_{pers} across different thresholds, while V_{com} and $V_{\text{com-dim}}$ remain close to their baseline levels with only minor fluctuations.

This observation is consistent with our main conclusion: RL-aligned models exhibit strong capability in the recognition stage, while the utilization stage remains the bottleneck.

Gemini 3 Flash Preview.

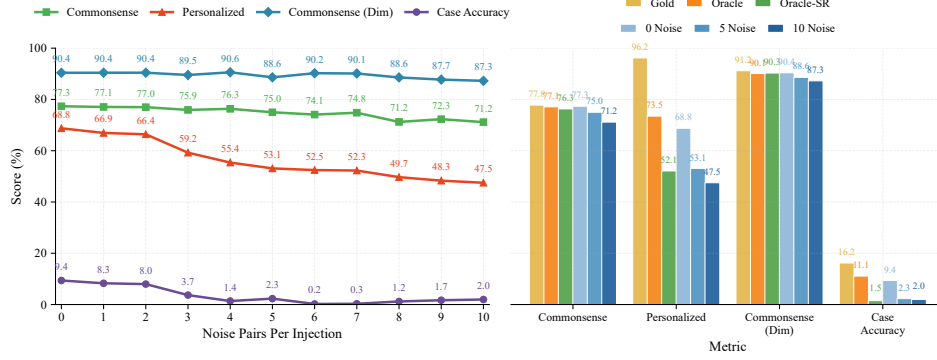


Figure 12: **Gemini 3 Flash Preview under noisy planning.** Left: Performance under increasing noise levels on DeepPlanning. Right: Comparison of the Gold, Oracle, and Oracle-SR settings.

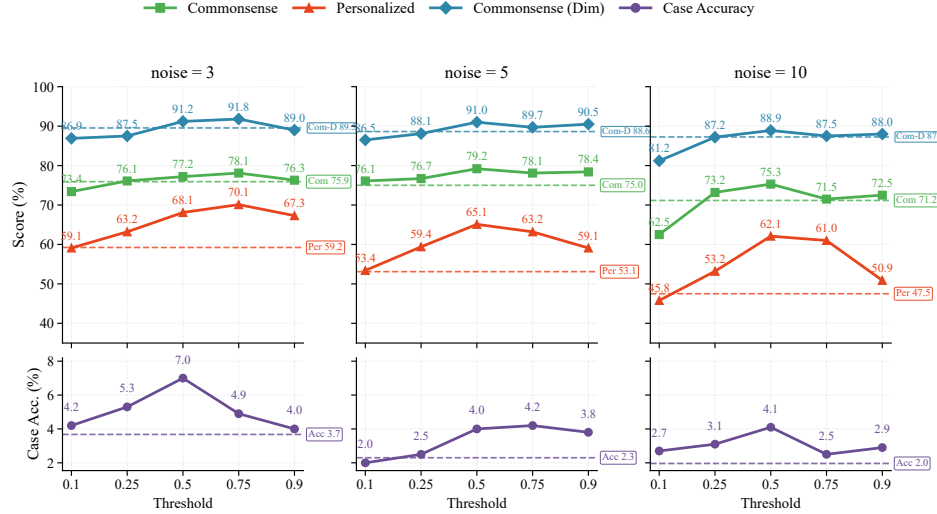


Figure 13: **Gemini 3 Flash Preview under different selection thresholds.** Results across noise levels with varying self-selection trigger thresholds.

The two figures above present the results of Gemini 3 Flash Preview. This model is RL-aligned and follows the same behavioral trend as Qwen3.5 reported in the main text.

As shown in Fig. 12, the overall performance is higher than Qwen3.5, yet the same degradation pattern under noise persists, indicating that the bottleneck remains unchanged despite stronger performance. In particular, V_{pers} still drops sharply under noise, while additional experiments settings (Gold, Oracle, and Oracle-SR) exhibit consistent recovery trends.

Fig. 13 further shows that, after introducing self-selection, both V_{pers} and CaseAcc improve more substantially compared to Qwen3.5. Meanwhile, V_{com} and $V_{\text{com-dim}}$ exhibit smaller fluctuations and remain more stable around their baseline levels.

Overall, these results reinforce our main conclusion: even for stronger RL-aligned models, the recognition stage is well preserved, while the utilization stage remains the primary bottleneck.

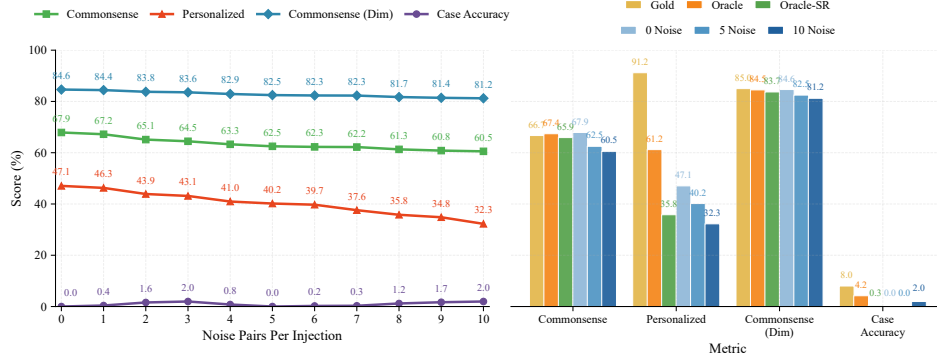
Kimi K2.5.

Figure 14: **Kimi K2.5 under noisy planning.** Left: Performance under increasing noise levels on DeepPlanning. Right: Comparison of the Gold, Oracle, and Oracle-SR settings.

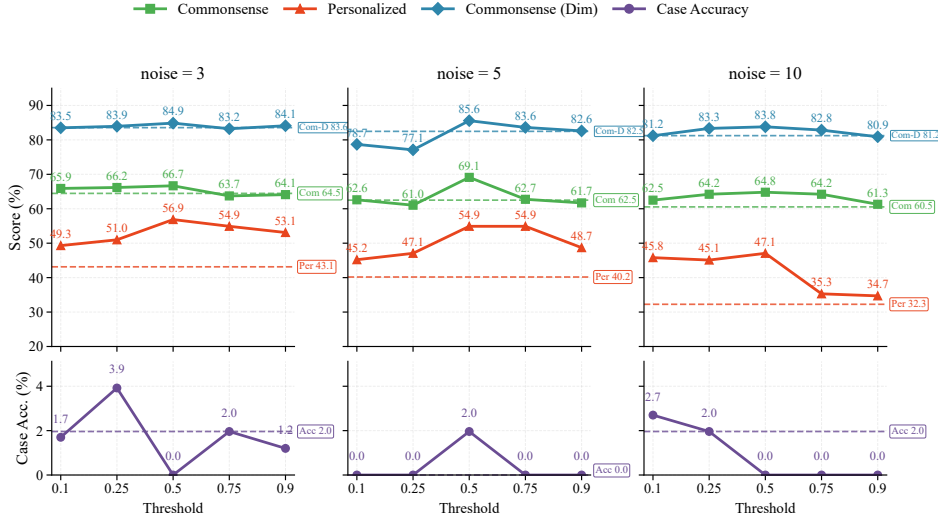


Figure 15: **Kimi K2.5 under different selection thresholds.** Results across noise levels with varying self-selection trigger thresholds.

The two figures above present the results of Kimi K2.5. This model is RL-aligned and exhibits behavior broadly consistent with Qwen3.5 in the main text.

As shown in Fig. 14, although the overall performance differs slightly from Qwen3.5, the trends under both noise injection and additional settings (*Gold*, *Oracle*, and *Oracle-SR*) remain similar. In particular, the degradation of V_{pers} under noise and its recovery in controlled settings follow the same pattern. Meanwhile, V_{com} remains relatively stable.

Fig. 15 further shows that, consistent with Qwen3.5, introducing self-selection leads to a clear improvement in V_{pers} across different thresholds, while V_{com} and $V_{\text{com-dim}}$ stay close to their baseline levels with only minor variations.

Overall, these results are consistent with our main conclusion: RL-aligned models demonstrate strong capability in the recognition stage, while the utilization stage remains the primary bottleneck.

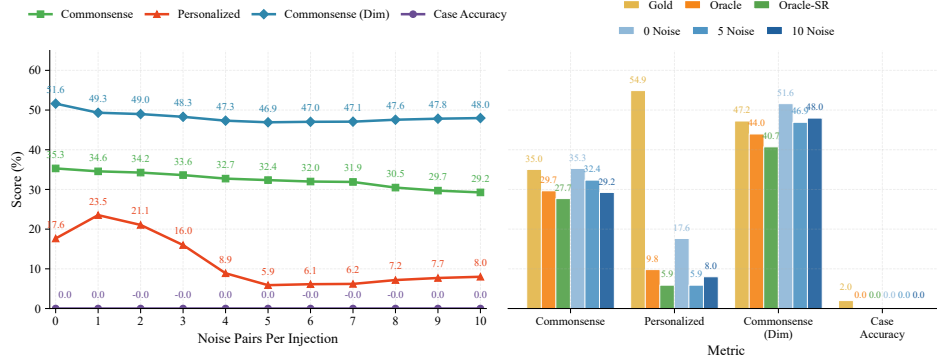
GPT-OSS-120B.

Figure 16: **GPT-OSS-120B under noisy planning.** Left: Performance under increasing noise levels on DeepPlanning. Right: Comparison of the Gold, Oracle, and Oracle-SR settings.

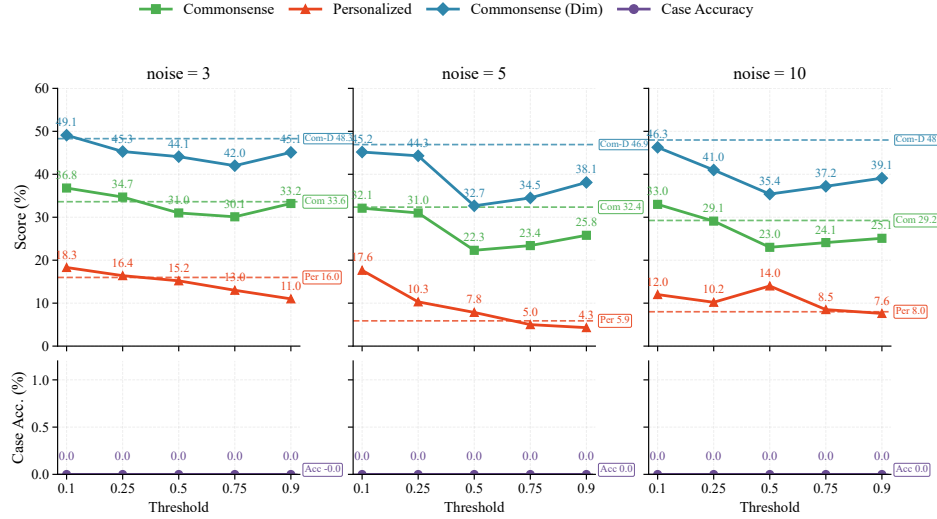


Figure 17: **GPT-OSS-120B under different selection thresholds.** Results across noise levels with varying self-selection trigger thresholds.

The two figures above present the results of GPT-OSS-120B. Unlike the models above, it is not RL-aligned and does not follow the same behavioral trend as Qwen3.5.

As shown in Fig. 16, although the overall performance is lower, both V_{com} and $V_{\text{com-dim}}$ exhibit a relatively mild and gradual decline as noise increases, indicating that basic commonsense reasoning remains stable under noise. In contrast, the behavior of V_{pers} is more nuanced: while it eventually collapses at higher noise levels, it shows slight improvements under low-noise conditions. This suggests that moderate noise may occasionally help highlight relevant information, potentially improving attention over key constraints.

Fig. 17 further reveals a different response to self-selection compared to RL-aligned models. While selection does not consistently improve V_{pers} , it leads to a noticeable degradation in both V_{com} and $V_{\text{com-dim}}$, indicating that filtering may remove useful contextual signals when recognition itself is unreliable.

Overall, these results highlight a key contrast: without agentic alignment, the model exhibits weaker and less stable recognition, and interventions such as selection can further disrupt performance rather than improve it.

D Additional Statistical Analysis

To further validate the recognition-utilization analysis, we conduct paired Wilcoxon signed-rank tests on the recognition-related metric ($V_{\text{com-dim}}$) and the constraint-adherence metric (V_{pers}). We report the mean difference, Cohen’s d , rank-biserial correlation, and p -value in Table 4. The results confirm that semantic noise significantly affects both stages, with a stronger impact on constraint adherence. Meanwhile, self-selection provides significant recovery on V_{pers} under both noise levels. $V_{\text{com-dim}}$ remains comparatively more stable, although it also exhibits measurable changes under severe noise, suggesting that recognition is affected but is less sensitive than constraint adherence.

Table 4: Additional statistical tests for recognition and utilization.

Test	Mean diff	Cohen’s d	Rank-biserial r	p -value
Vpers: 0 Noise vs 5 Noise	0.137	0.26	0.47	0.071
Vpers: 0 Noise vs 10 Noise	0.314	0.51	0.67	0.001
Vpers: 5 Noise vs 10 Noise	0.176	0.30	0.47	0.039
Vpers: 5 Noise vs 5 Noise + self-selection	-0.216	-0.37	-0.58	0.012
Vpers: 10 Noise vs 10 Noise + self-selection	-0.275	-0.41	-0.54	0.006
Vcom-dim: 0 Noise vs 10 Noise	0.071	0.64	0.63	0.0001

We additionally compute bootstrap confidence intervals for the trajectory centroids in Figure 7. Under 0 Noise, the centroid is ($V_{\text{com-dim}} = 87.0 [84.7, 89.2]$, $V_{\text{pers}} = 64.7 [51.0, 78.4]$). Under 10 Noise, it shifts to ($79.8 [77.2, 82.4]$, $33.3 [21.6, 47.1]$), while 10 Noise with self-selection recovers to ($82.7 [79.9, 85.2]$, $60.8 [47.1, 74.5]$). These confidence intervals further support that noise primarily affects constraint adherence and that self-selection substantially restores performance.

E Case Studies

E.1 High commonsense but low personalization (Com: 1.0 / Pers: 0.0)

Case type: high commonsense, low personalization.

Evaluation signals.

- **commonsense_score** = 1.0
- **personalized_score** = 0.0

All commonsense dimensions (e.g., route consistency, time feasibility, itinerary structure, and cost calculation) are fully satisfied.

Observed plan quality. The generated itinerary is coherent, executable, and well-structured:

- consistent travel sequence with valid timing
- reasonable arrangement of attractions and meals
- correct budget estimation

Failure source. A fine-grained user constraint is not satisfied:

- required: **Western fusion restaurant near Mulong Lake**
- selected: **Ruozhuo Western Restaurant**

Interpretation. This case highlights a fundamental gap: commonsense planning quality and personalized constraint satisfaction are decoupled. A plan can be globally coherent and fully feasible, yet still fail to meet specific user requirements.

E.2 Selection restores the decisive anchor (0 → 1)

Case type: successful recovery after selection.

Gold constraints. The plan should use the direct train pair G1302 → G1301 and select the highest-rated hotel, *Hanting Shanghai Jiaoda Jiangchuan Rd. Metro Hotel*.

Observed outputs.

- **Baseline:** the correct train backbone is already identified, but the plan chooses the wrong hotel, *QiuZhu Hotel Shanghai*, which leads to failures on *hotel_highest_rated* and *budget_constraint*.
- **After selection:** the plan switches to *Hanting Shanghai Jiaoda Jiangchuan Rd. Metro Hotel*, and all hard constraints pass.

Decisive evidence retained after selection.

- *Hanting Shanghai Jiaoda Jiangchuan Rd. Metro Hotel* — price: 275, score: 5.0
- G1302 — remaining seats: 7
- G1301 — remaining seats: 8

Interpretation. Selection repairs the plan not by changing transport retrieval, but by preserving the correct *anchor entity* strongly enough for final synthesis.

E.3 Selection preserves evidence but not decision (0 → 0)

Case type: persistent failure after selection.

Gold constraints. The plan should use the cheapest direct outbound train, the highest-rated 3-star hotel, and satisfy two restaurant constraints, including a *Western Fusion Restaurant* near Mulong Lake.

Observed outputs.

- **Baseline:** the main transport and hotel choices are correct, but the final dinner is still *Ruozhuo Western Restaurant*, which violates *restaurant_specific_cuisine_nearby*.
- **After selection:** the same incorrect restaurant choice remains, and the final evaluation is unchanged.

Key point. Selection preserves the relevant evidence but does not guarantee correct downstream constraint execution, as the planner may still repeat the same mistake.

E.4 Selection introduces attribute drift (1 → 0)

Case type: degradation after selection.

Gold constraints. The plan requires the cheapest direct flight, a 3-star hotel with pool, all required natural-attraction visits, and the top-rated historical attraction.

Observed outputs.

- **Baseline:** the plan correctly uses All Seasons Quanzhou West Street Zhongshan Rd. Hotel, and all hard constraints pass.
- **After selection:** the planner switches to Orange Hotel Quanzhou Jinjiang Airport Yangguang Rd., and the final evaluation fails on hotel.star.service.required.

Conflicting retained evidence after selection.

- All Seasons Quanzhou West Street Zhongshan Rd. Hotel with services including pool
- A conflicting retained version of the same hotel with incomplete service attributes
- Orange Hotel Quanzhou Jinjiang Airport Yangguang Rd. remains as a competing alternative

Interpretation. Selection may destabilize key attributes in memory, causing the planner to shift away from a previously correct anchor and leading to degraded outcomes.

F Limitation

Our experiments are conducted on a limited set of models due to resource constraints. While we include several representative and relatively recent models across different providers, including both open-weight and non-open frontier systems, we do not perform a large-scale evaluation over a broader model spectrum. As such, the empirical findings may not fully generalize to all model families. Broader closed-model sweeps remain costly for full trajectory-level, noise-controlled agent experiments.

In addition, our experiments use DeepPlanning as the main controlled testbed because it provides multi-tool interaction, dynamic observations, hard user constraints, and constraint-level evaluation. However, it remains one controlled domain, and future work should extend the analysis to other agent settings, such as open-ended search and coding agents.

Although we observe consistent behavioral patterns among models that have undergone agentic alignment (e.g., RL-based optimization), we cannot conclusively attribute these differences solely to the presence of such alignment. The observed variations may also be influenced by intrinsic model capabilities, architectural differences, or training data. Therefore, the comparisons across models should not be interpreted as strictly controlled or fully fair.

Nevertheless, we include GPT-OSS-120B as a representative model without explicit agentic alignment to provide a contrasting case. However, this choice is limited by model availability and resource constraints and may not fully capture the diversity of non-agentially aligned models.