Can Large Language Models Tell When a Problem Is Underspecified?

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated impressive reasoning capabilities on benchmarks that implicitly assume information completeness. However, real-world decision-making often involves underspecified queries where the correct response is to abstain or request clarification rather than force a solution. While traditional task-oriented dialogue systems address this via explicit slot filling, general-purpose LLMs lack such mechanisms, often hallucinating details to satisfy a "compulsion to respond." Current evaluations fail to detect this failure mode, as existing backward reasoning studies typically condition models on known answers, introducing ground-truth leakage that masks the inability to assess solvability ab initio. To bridge this gap, we introduce a specialized benchmark for Missing Information Detection, evaluating whether models can identify necessary prerequisites solely from questions without answer access. We further propose Reverse Thinking for Information Completeness Assessment (RT-ICA), a goal-directed reasoning framework that guides models to infer prerequisites from the implicit solution goal. Experimental results on our curated datasets reveal that while standard models struggle to detect incompleteness, RT-ICA significantly improves reliability by operationalizing a "reasoning-level slot filling" mechanism. Our work highlights a critical blind spot in reasoning evaluation and offers a robust methodology for safer deployment.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide spectrum of tasks, including complex reasoning, mathematical problem solving, and question answering (Brown et al., 2020; OpenAI et al., 2023). In these standard evaluations, tasks are typically framed within idealized settings where all necessary information is explicitly provided and a unique solution exists. Consequently, models are assessed primarily on their ability to derive correct answers via forward reasoning, often enhanced by structured prompting strategies such as Chain-of-Thought (CoT) (Wei et al., 2022) and Tree-of-Thought (ToT) (Yao et al., 2024). However, this evaluation paradigm implicitly assumes that every input is complete and solvable. By focusing exclusively on solution generation, current benchmarks fail to measure a model's ability to validate problem premises, creating a blind spot in evaluating how models behave when faced with unsolvable or underspecified queries.

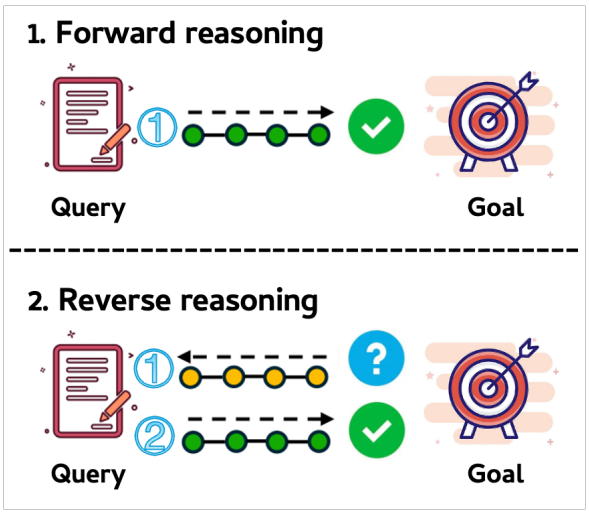

Figure 1: Conceptual comparison. **Top:** Forward reasoning chains premises to a solution, implicitly assuming the input is complete. **Bottom:** Reverse thinking works backward from the goal to verify prerequisites. This enables the detection of missing information (Question Mark) when necessary conditions are not satisfied, preventing unjustified conclusions.

This limitation is critical because real-world scenarios frequently involve underspecified or incomplete problem descriptions. While task-oriented dialogue systems address this ambiguity via explicit slot filling based on fixed schemas (Young

et al., 2013), general-purpose LLMs operate without such predefined constraints. In open-domain contexts, such as complex planning (Geffner and Bonet, 2013) and logical inference (Russell and Norvig, 2010), the model must implicitly perform what we term "reasoning-level slot filling": determining whether the logical premises required for a conclusion are actually present. The absence of this verification step is the root cause of many reliability issues: when models trained on complete data encounter missing prerequisites in practice, they often hallucinate details to force a solution. Consequently, a model that excels on standard leaderboards may prove unreliable in deployment, as it lacks the capability to detect invalid questions and abstain from answering.

Yet, this critical deficiency remains invisible under current evaluation protocols. As illustrated in Figure 1, standard benchmarks rely on forward reasoning (top), where models chain given premises to derive a solution. This paradigm implicitly assumes information completeness. In contrast, reverse thinking (bottom) offers a mechanism to verify solvability by working backward from the goal to its prerequisites. While recent works have explored such backward mechanisms (Deb et al., 2024; Chen et al., 2024; Xu et al., 2025; Ranaldi and Pucci, 2025), they suffer from a fundamental limitation: they typically condition the model on a known answer. In real-world settings, the answer is unknown; providing it allows the model to reverse-engineer missing information via ground-truth leakage, rather than detecting insufficiency from the question itself. Thus, there is an urgent need for a benchmark that strictly evaluates this prerequisite-checking capability ab initio, orthogonal to standard solution generation.

To address this systematic oversight, we introduce a benchmark designed to evaluate LLMs under a stricter, answer-agnostic constraint: the model must determine solvability and identify specific missing prerequisites based solely on the question. This formulation explicitly prevents the answer-driven inference strategies criticized above, isolating the model's capability to assess problem completeness before attempting a solution. To operationalize this evaluation, we propose Reverse Thinking for Information Completeness Assessment (RT-ICA) as a unifying reasoning framework. Unlike previous backward reasoning approaches that hinge on known outcomes, RT-ICA guides models to reason backward from the implicit solution goal. By enumerating necessary prerequisites and verifying their satisfaction in the context (as shown in Figure 2), RT-ICA effectively implements the "reasoning-level slot filling" mechanism required for robust real-world deployment.

Our contributions are summarized as follows:

- We identify a systematic blind spot in current reasoning evaluations and formalize the task of *Missing Information Detection*. Unlike existing paradigms, our benchmark evaluates the model's capability to assess solvability and identify missing prerequisites under strict, answer-agnostic constraints, thereby preventing ground-truth leakage.

- We construct two curated datasets designed to rigorously test this capability across varying complexity levels. These datasets capture the structural diversity of real-world incompleteness, ranging from direct premise gaps to subtle, multi-step logical deficiencies.

- We propose *RT-ICA*, a goal-directed reverse reasoning framework. By inferring necessary conditions from the *problem goal* rather than a known answer, RT-ICA effectively operationalizes "reasoning-level slot filling," enabling models to verify information completeness before generation.

2 Related Work

2.1 Forward Reasoning Paradigms

A large body of prior work has investigated the reasoning capabilities of large language models through *forward reasoning* paradigms, where models derive answers from fully specified inputs. Chain-of-Thought (CoT) prompting enables step-by-step reasoning and has shown consistent improvements across mathematical, logical, and commonsense tasks (Kojima et al., 2022; Wei et al., 2022). Subsequent extensions, such as self-consistency, further enhance robustness by aggregating multiple reasoning trajectories (Wang et al., 2023).

Beyond linear reasoning, more structured approaches explicitly explore the reasoning space. Tree-of-Thought (ToT) performs deliberative search over multiple reasoning branches (Yao et al., 2024), while ReAct integrates reasoning with action and tool use (Yao et al., 2023). Other lines of

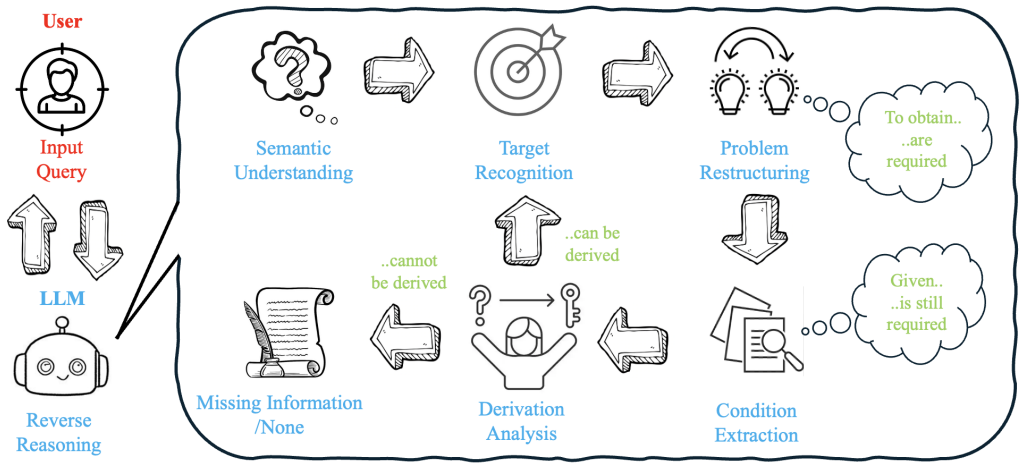

Figure 2: The overall framework of our proposed reverse thinking approach for missing information detection.

work incorporate self-reflection and critique mechanisms (Shinn et al., 2024; Madaan et al., 2024), as well as multi-agent collaboration frameworks that distribute reasoning across interacting agents (Du et al., 2024; Liang et al., 2023; Chen et al., 2023).

2.2 Backward Reasoning

Recent work has explored backward or reverse reasoning in LLMs, examining whether models can reconstruct premises, infer latent variables, or recover original questions from conclusions or answers (Deb et al., 2024; Chen et al., 2024; Xu et al., 2025; Ranaldi and Pucci, 2025). These studies highlight that backward inference exhibits reasoning behaviors and challenges distinct from standard forward reasoning. However, existing backward reasoning benchmarks are predominantly answer-conditioned: the correct outcome is explicitly provided to the model. This enables reverse inference by leveraging information embedded in the answer itself, and therefore does not directly test whether a model can determine problem solvability when the answer is unknown. As a result, performance in such settings is insufficient to assess missing information detection under answer-free conditions.

2.3 Connections to Abductive Reasoning

Abductive reasoning concerns inferring latent premises that best explain observed outcomes and has been widely studied in artificial intelligence and natural language understanding. In NLP, abductive inference has been explored in tasks such as commonsense reasoning and narrative understanding, where models hypothesize implicit assumptions to render observations coherent (Bhagavatula et al., 2020).

In mathematical and logical problem solving, backward and abductive reasoning have been used to identify missing premises required for solution derivation (Qin et al., 2020, 2022). These approaches are typically developed in symbolic or highly structured environments, where reasoning rules and variable dependencies are explicitly defined.

3 Dataset Construction

A core failure mode of LLMs in practice is their tendency to produce answers even when problems are underspecified, highlighting a critical gap in reasoning robustness. However, this capability is rarely evaluated, as existing benchmarks predominantly assume information-complete inputs. To address this, we construct two dedicated benchmark datasets for *missing information detection*. Our datasets are derived from well-established mathematical reasoning benchmarks and systematically modified to include both complete and incomplete problem instances. The construction process emphasizes controllability, interpretability, and annotation consistency. The datasets and accompanying documentation are publicly available at an anonymous repository:[1]

3.1 Problem Generation

We construct our datasets based on two widely used multi-step reasoning benchmarks: *GSM8K* (Cobbe et al., 2021) and *Math500* (mat, 2024). GSM8K-derived problems provide simpler cases with shal-

[1] https://anonymous.4open.science/r/RT-ICA-Data-DB2D/

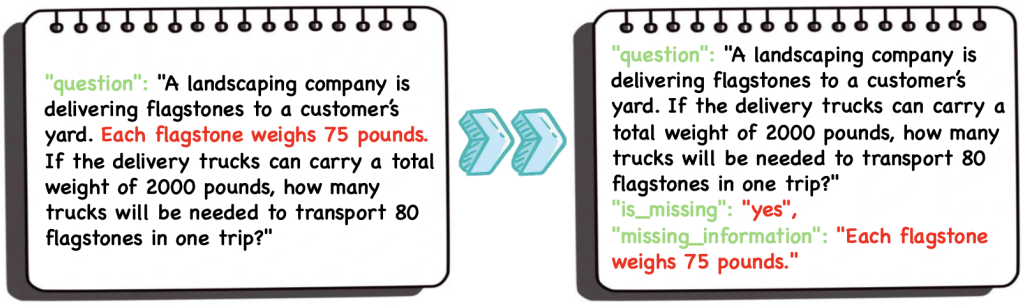

Figure 3: Conceptual diagram of dataset construction items.

low prerequisite dependencies, while Math500-derived problems involve deeper reasoning chains and more complex indirect missing conditions. From each benchmark, we retain a subset of original problems as complete instances, serving as negative examples.

Incomplete instances are created via a controlled human annotation process: annotators first identify the minimal *goal prerequisites* required to solve a problem, including numerical values, variable definitions, relational constraints, or implicit assumptions, and then selectively remove one or more elements. Each modified problem remains meaningful but becomes underspecified, and the removed information can be explicitly articulated as a concrete prerequisite.

3.2 Annotation Schema

Each data instance is annotated using a structured schema consisting of three fields:

- **question**: the problem statement presented to the model;
- **is_missing**: a binary label indicating whether the problem contains missing information (yes or no);
- **missing_information**: a textual description of the missing prerequisite(s), provided only when is_missing = yes.

For complete problems, the missing_information field is left empty. Annotators are instructed to describe missing elements concisely and concretely, avoiding solution-oriented reasoning or speculative explanations.

3.3 Quality Control

To ensure annotation reliability, each incomplete instance is independently reviewed by multiple annotators. Disagreements are resolved through discussion, with priority given to interpretations that preserve the minimality and necessity of the missing condition. This process reduces subjectivity and enforces consistency across the dataset.

3.4 Dataset Example

We construct two evaluation datasets with balanced distributions of complete and incomplete problems. *test_gsm8k* contains 505 problems, including 234 incomplete and 271 complete instances, while *test_math* consists of 501 problems, with 229 incomplete and 272 complete instances. This design ensures realistic assessment of models' ability to detect missing information without introducing artificial ambiguity.

Figure 3 presents representative examples from our dataset. The example explicitly separates the original question, the binary completeness label, and the annotated missing information, highlighting the distinction between solvable and underspecified problem formulations. Together, the balanced category design and structured annotation provide a **comprehensive benchmark** for evaluating model competency in information completeness assessment.

4 Method

4.1 Problem Formulation

We study missing information detection in incomplete question answering. Given a question Q that may be underspecified, the task is to determine whether the information provided is sufficient to support a valid solution and, if not, to explicitly identify what information is missing.

Formally, let A denote the set of information explicitly stated in the question Q, including variables, values, and constraints. Let G denote the (implicit) solution goal that the question intends to resolve. The output of the task consists of (1) a binary indicator $I_{\text{missing}} \in \{\text{yes}, \text{no}\}$, and (2) a set of missing information elements $M = \{m_1, \ldots, m_k\}$

when $I_{\text{missing}} = \text{yes}$. Each m_i corresponds to an atomic prerequisite—such as an undefined quantity, an absent numerical value, or a missing relational constraint—that is necessary to derive G.

Missing information detection can be formalized as assessing whether the available information A alone is sufficient to entail the goal G. If not, the task is to identify a set M such that:

$$A + M \rightarrow G,$$

Importantly, the objective is not to compute the final answer, but to evaluate information sufficiency and diagnose which prerequisites are absent.

Most existing reasoning paradigms do not fit this setting. Forward reasoning assumes known A and G, while backward reasoning assumes a known G. Here, only A is observed and both G and M are implicit, requiring the model to infer the goal and identify missing prerequisites. Although crucial for real-world LLM use with underspecified queries, this capability is rarely evaluated, with little benchmark data available—motivating our dataset and RT-ICA framework.

The complexity of the problem. Missing information detection requires reasoning over implicit solution goals and their prerequisite structures, which is inherently more complex than standard forward answer derivation. Unlike conventional paradigms that assume either a known goal or complete premises, this setting demands iterative backward exploration and verification of prerequisite dependencies. As a result, both the time and space complexity of such reasoning grow with the depth and branching of prerequisite structures. We provide a formal complexity analysis of RT-ICA, including worst-case time and space bounds, in Appendix A.

4.2 RT-ICA Framework

We propose **Reverse Thinking for Information Completeness Assessment (RT-ICA)**, a goal-directed framework that detects missing information under answer-free settings by performing backward reasoning from the implicit solution goal. RT-ICA comprises three tightly coupled modules, corresponding to the operations in Algorithm 1, which collectively identify, extract, and verify the prerequisites required to resolve a question.

Semantic Understanding: Target Recognition. This module identifies the implicit solution objective from a given question, separating the provided information from unknowns. During this process, it also generates the initial set of *goal prerequisites*, which serve as the starting point for subsequent condition extraction and verification. Target recognition thus establishes both the reasoning goal and the high-level structure of prerequisites for the question.

Reverse Reasoning: Problem Restructuring & Condition Extraction. The core module performs *condition extraction* to identify all explicit and implicit information present in the question. Each extracted condition represents a prerequisite that may be necessary to satisfy one of the goal prerequisites. When a condition itself depends on further prerequisites, *enumerate prerequisites* recursively expands the dependency structure until all atomic conditions are identified. This forms the complete set of conditions that need to be verified against the question.

Evaluation: Derivation Analysis & Missing Information Identification. Each condition in the expanded set is iteratively checked for availability using a *check availability* procedure, which determines whether it is explicitly stated in the question or implicitly derivable from other verified conditions. Conditions that cannot be verified are collected into the *missing information set*, and a corresponding *missing indicator* is generated to signal whether the question is incomplete. If no missing conditions remain, the question is considered information-complete.

By explicitly incorporating *goal prerequisites*, *condition extraction*, *prerequisite enumeration*, and iterative *availability checking*, RT-ICA operationalizes a systematic reasoning-level slot filling process. This enables interpretable assessments of information sufficiency without relying on answer-driven inference. Through this process, RT-ICA first deduces the implicit goal G from the given context A, then pinpoints the missing prerequisites M that preclude its solution, thereby structuring the path $A \rightarrow (G, M)$ without relying on a known answer.

Algorithm 1 Missing Information Detection via Iterative Reverse Thinking

Require: Question Q, Language Model $\mathcal{M}$
Ensure: Missing indicator I_{missing}, Missing information set M

```
K ← ConditionExtraction(Q)
M ← ∅
C ← GoalPrerequisites(Q, M)
while C ≠ ∅ do
    Select and remove a condition c from C
    if CheckAvailability(c, K) then
        continue
    end if
    P_c ← EnumeratePrerequisites(c, M)
    if P_c = ∅ then
        M ← M ∪ {c}
    else
        C ← (C \ {c}) ∪ P_c
    end if
end while
if M ≠ ∅ then
    I_missing ← yes
else
    I_missing ← no
end if
return I_missing, M
```

5 Experiment

To evaluate the effectiveness of our proposed RT-ICA method, we conducted comprehensive experiments on two distinct datasets: *test_gsm8k* and *test_math*. We compared the performance of several baseline models against their enhanced versions integrated with RT-ICA, as well as other adapted methods, such as Rephrase, CYW, and RFF. The models under investigation include GPT-5.1 (OpenAI, 2025), GPT-3.5-turbo (OpenAI, 2023), and DeepSeek-V3.2-Exp (DeepSeek-AI, 2025).

Baselines. We compare RT-ICA with several established reasoning paradigms adapted to the task of detecting missing information:

- **Standard IO**: The base model directly determines whether the problem contains missing information, without utilizing any intermediate reasoning steps or structured decomposition.
- **Rephrase** (Deb et al., 2024): The model reformulates the problem by marking unknown quantities with variables (e.g., 'x') and converting it into an equation-solving format. This method assumes that a solvable equation indicates completeness.
- **Check Your Work (CYW)** (Deb et al., 2024): The model hypothesizes missing values, substitutes them into the problem, solves it, and checks for logical consistency. Persistent inconsistency suggests the problem is underspecified.
- **Reason from Future (RFF)** (Xu et al., 2025): This approach alternates between reverse planning and forward execution to identify missing conditions. If a condition cannot be satisfied from the available context, it is flagged as missing information.

Each baseline has been carefully adapted to align with our task's objective. All experiments are conducted under consistent conditions, ensuring fair comparison with RT-ICA.

Evaluation Protocol. We evaluated the models using three complementary metrics: overall accuracy across all problems, accuracy on incomplete problems where information is intentionally omitted, and accuracy on complete problems containing all necessary components. This protocol captures the model's ability to detect missing information while maintaining reliability on fully specified questions.

Model Evaluation. RT-ICA was evaluated on three state-of-the-art models: GPT-5.1 (OpenAI, 2025), GPT-3.5-turbo (OpenAI, 2023), and DeepSeek-V3.2-Exp (DeepSeek-AI, 2025). All experiments used consistent parameters (temperature = 0.7, top-p = 0.9), and each configuration was evaluated across three random seeds for statistical reliability. The RT-ICA prompt architecture employed structured templates to guide models through reverse reasoning.

5.1 Results

Table 1 summarizes the performance of RT-ICA against baselines. A key finding is that RT-ICA consistently achieves superior overall accuracy while markedly improving detection of incomplete problems (the *Yes* category).

Methods like Standard IO and Rephrase, which rely on single forward passes, fail to audit informational sufficiency. More sophisticated iterative methods like CYW exhibit a critical limitation:

Table 1: Performance comparison of baseline models and RT-ICA enhanced models on two datasets. All values are accuracy percentages.

Model	Dataset	Overall (%)	Yes Cat. (%)	No Cat. (%)	Min(%)	Tokens
GPT-3.5-turbo	test_gsm8k	44.76	30.77	58.49	30.77	145
+Rephrase	test_gsm8k	49.70	52.14	47.60	47.60	288
+CYW	test_gsm8k	60.99	19.66	**96.68**	19.66	219
+RFF	test_gsm8k	**65.15**	35.47	90.77	35.47	699
+RT-ICA	test_gsm8k	60.99	**71.37**	52.03	**52.03**	567
GPT-3.5-turbo	test_math	38.92	19.65	55.15	19.65	160
+Rephrase	test_math	44.91	29.26	58.09	29.26	311
+CYW	test_math	53.89	3.93	95.96	3.93	310
+RFF	test_math	56.69	30.13	**79.04**	30.13	689
+RT-ICA	test_math	**57.29**	**35.81**	75.37	**35.81**	654
DS-V3.2-Exp	test_gsm8k	78.21	67.52	87.45	67.52	110
+Rephrase	test_gsm8k	78.61	**87.61**	70.85	70.85	201
+CYW	test_gsm8k	83.56	79.91	86.72	79.91	313
+RFF	test_gsm8k	86.34	76.92	**94.46**	76.92	635
+RT-ICA	test_gsm8k	**86.73**	80.34	92.25	**80.34**	588
DS-V3.2-Exp	test_math	57.49	18.34	90.44	28.34	177
+Rephrase	test_math	56.29	35.81	73.53	35.81	264
+CYW	test_math	59.68	16.16	**96.85**	16.16	323
+RFF	test_math	62.97	41.05	81.25	41.05	732
+RT-ICA	test_math	**65.07**	**43.67**	83.09	**43.67**	597
GPT-5.1	test_gsm8k	79.21	70.51	86.72	70.51	190
+Rephrase	test_gsm8k	77.62	80.77	74.91	74.91	334
+CYW	test_gsm8k	79.80	69.66	88.56	69.66	297
+RFF	test_gsm8k	**85.74**	72.65	**97.05**	72.65	681
+RT-ICA	test_gsm8k	85.54	**77.35**	92.62	**77.35**	741
GPT-5.1	test_math	60.08	17.90	**95.59**	17.90	211
+Rephrase	test_math	60.28	39.30	77.94	39.30	344
+CYW	test_math	57.29	20.96	87.87	20.29	401
+RFF	test_math	64.27	**51.09**	75.37	**51.09**	821
+RT-ICA	test_math	**66.47**	48.47	81.62	48.47	799

Table 2: Ablation study analyzing component contributions in RT-ICA framework. Evaluated on test_math dataset (GPT-3.5-turbo).

Component Variation	Overall	Yes Cat.	No Cat.
Full RT-ICA Framework	**57.29**	**35.81**	**75.37**
Semantic Understanding Module			
– w/o Target Recognition	45.99	12.98	73.78
Reverse Reasoning Core			
– w/o Problem Restructuring	51.39	33.59	66.38
– w/o Condition Extraction	55.35	31.28	75.61
– Forward Reasoning Only	40.10	20.51	56.59
Evaluation Module			
– w/o Derivation Analysis	53.37	29.72	73.28

they require hypothesizing a value for a missing element to initiate verification. When both the final answer *and* intermediate conditions are unknown—a common scenario in underspecified problems—this hypothesis lacks a foundation, causing the verification loop to fail. This explains the observed bias: high accuracy on complete problems but catastrophically low accuracy on incomplete ones.

In contrast, RT-ICA addresses this core challenge by not assuming any hypothetical values. It employs goal-directed reverse reasoning to first deduce the necessary conditions for a solution, then verifies each against the context. This approach uniquely diagnoses gaps even when no numerical anchor is provided, because it reasons about the structure of required information rather than its instantiation. Consequently, RT-ICA delivers balanced performance, as evidenced by its high minimum accuracy.

This structural advantage is evident across datasets. On *test_gsm8k*, RT-ICA dramatically lifts *Yes*-category accuracy (e.g., from 30.77% to 71.37% for GPT-3.5-turbo). On the more complex *test_math*, it sustains superior performance, identifying more underspecified problems.

Computationally, RT-ICA incurs moderate overhead but remains token-efficient against multi-step paradigms like RFF. This cost is justified by its fundamental capability to reliably detect missing information—a capability forward or verification-based methods lack.

In summary, reverse thinking provides a principled mechanism for information completeness assessment. By identifying unsatisfied prerequisites without hypothetical completions, RT-ICA offers a robust solution for a critical weakness in current LLM reasoning.

5.2 Ablation Study

Table 2 presents an ablation study on the *test_math* dataset using GPT-3.5-turbo, analyzing the contribution of individual components in the RT-ICA framework.

Semantic Understanding Stage. Removing target recognition causes the most severe performance degradation, with overall accuracy dropping by 11.3 percentage points and *Yes* category accuracy collapsing from 35.81% to 12.98%. This confirms that precise identification of the solution objective is the critical first step, without which subsequent reverse reasoning lacks direction.

Reverse Reasoning Core. The components enabling systematic backward reasoning are fundamental. Omitting problem restructuring reduces overall accuracy by 5.9 points, while removing condition extraction has a milder impact. Most tellingly, replacing the entire reverse core with standard forward reasoning causes a catastrophic drop: *Yes* category accuracy falls to 20.51% (vs. 35.81%) and overall accuracy plummets by 17.19 points. This stark contrast validates that reverse, goal-directed reasoning is essential for detecting what is missing.

Evaluation Module. Removing the derivation analysis step, which checks if prerequisites can be logically satisfied, reduces overall accuracy by 3.92 points and harms *Yes* category performance by 6.09 points. This underscores that merely listing conditions is insufficient; explicit verification of their derivability is necessary for reliable diagnosis.

Overall, the ablation confirms that RT-ICA's effectiveness stems from the synergistic integration of its components. Semantic understanding provides accurate grounding, the reverse reasoning core enables systematic prerequisite discovery, and the evaluation module ensures robust verification. The framework fails decisively when reverting to forward reasoning, highlighting the paradigm shift at its core.

6 Conclusion

In this work, we study missing information detection as a standalone reasoning task, where models must assess information sufficiency without access to the correct answer. To support this setting, we introduce a benchmark and two difficulty-controlled datasets that explicitly distinguish complete and underspecified problem formulations, reflecting realistic usage scenarios where inputs are often incomplete.

Building on this formulation, we propose RT-ICA, a reverse thinking framework that reasons backward from the solution goal to enumerate and verify prerequisite conditions. Experiments across multiple models and datasets show that RT-ICA consistently improves missing information detection accuracy, particularly for incomplete problems, demonstrating the effectiveness of reverse thinking under answer-free constraints.

We hope this work encourages further investigation into information completeness assessment as a complementary dimension of reasoning evaluation, alongside traditional forward solution derivation.

Limitations

A key limitation of the proposed benchmark lies in the inherent difficulty of defining and annotating missing information for problems with deep or highly interdependent prerequisite structures. In such cases, determining the minimal set of missing conditions may require multiple layers of implicit derivation, introducing potential ambiguity even for human annotators.

This challenge reflects a broader open problem in information completeness assessment rather than a limitation of a specific method. Future work may explore richer annotation schemes or hierarchical representations of prerequisites to better capture complex dependency structures.

References

2024. Math500: A test set for measuring mathematical reasoning of large language models. Available at https://huggingface.co/datasets/ankner/math-500.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Justin Chih-Yao Chen, Zifeng Wang, Hamid Palangi, Rujun Han, Sayna Ebrahimi, Long T Le, Vincent Perot, Swaroop Mishra, Mohit Bansal, Chen-Yu Lee, and Tomas Pfister. 2024. Reverse thinking makes llms stronger reasoners. *arXiv preprint arXiv:2411.19865*.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yujia Qin, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. *arXiv preprint*, arXiv:2308.10848.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Guillaume Lample, Wojciech Zaremba, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Aniruddha Deb, Neeva Oza, Sarthak Singla, Dinesh Khandelwal, Dinesh Garg, and Parag Singla. 2024. Fill in the blank: Exploring and enhancing llm capabilities for backward reasoning in math word problems. *arXiv preprint arXiv:2310.01991*.

DeepSeek-AI. 2025. Deepseek-v3.2-exp: Boosting long-context efficiency with deepseek sparse attention.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 11733–11763. PMLR.

Hector Geffner and Blai Bonet. 2013. A concise introduction to models and methods for automated planning. In *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool Publishers.

Takeshi Kojima, Shixiang Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *NeurIPS*.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint*, arXiv:2305.19118.

Aman Madaan and 1 others. 2024. Self-refine: Iterative refinement with self-feedback. In *ICLR*.

OpenAI. 2023. Gpt-3.5 technical report. https://platform.openai.com/docs/guides/gpt. Accessed: 2025-08-02.

OpenAI. 2025. Gpt-5.1 system card. https://openai.com. Accessed: 2025-10-02.

Josh Achiam OpenAI, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D. Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 794–805.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. Cold decoding: Energy-based constrained text generation with langevin dynamics. In *Advances in Neural Information Processing Systems*, volume 35, pages 9538–9551.

Leonardo Ranaldi and Giulia Pucci. 2025. Exploring backward reasoning in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, page 366.

Stuart Russell and Peter Norvig. 2010. *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Noah Shinn, Becky Labash, Ashwin Gopinath, and 1 others. 2024. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, and Ed Chi. 2023. Self-consistency improves chain of thought reasoning in language models. In *ICLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Yinlong Xu, Yanzhao Zheng, Shuoshuo Sun, Shuaihan Huang, Baohua Dong, Hangcheng Zhu, Ruohui Huang, Gang Yu, Hongxia Xu, and Jian Wu. 2025. Reason from future: Reverse thought chain enhances llm reasoning. *arXiv preprint arXiv:2506.03673*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR) Workshops*. ArXiv:2210.03629.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

A Appendix A: Theoretical Foundations and Mathematical Formulations

Formal Problem Definition

We provide the complete mathematical formalization of the missing information detection problem and the RT-ICA framework.

Problem Space Definition

Let $\mathcal{Q}$ denote the problem space, where each problem $q \in \mathcal{Q}$ is represented as a tuple:

$$q = (\mathcal{V}, \mathcal{R}, \mathcal{C}, \mathcal{O}) \tag{1}$$

where:

- $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of variables
- $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$ is the set of relations and constraints
- $\mathcal{C} \subseteq \mathcal{V} \times \mathbb{R}$ is the set of known conditions (variable-value assignments)
- $\mathcal{O} : \mathcal{V} \to \mathbb{R}$ is the objective function to be computed

A problem q is considered *complete* if there exists a unique solution path from $\mathcal{C}$ to $\mathcal{O}$:

$$\exists!\ \pi : \mathcal{C} \to \mathcal{O} \text{ such that } \pi \vdash \mathcal{O} \tag{2}$$

The missing information detection task can be formulated as determining whether:

$$\mathcal{M}(q) = \{\phi \in \Phi \mid \pi \nvdash \phi \text{ and } \phi \text{ is necessary for } \mathcal{O}\} \tag{3}$$

where Φ represents the complete set of prerequisites for solving q.

Complete Algorithm Specification

Algorithm 2 Complete Reverse Thinking for Information Completeness Assessment (RT-ICA)

Require: Problem q, Language Model $\mathcal{M}$, Threshold θ
Ensure: Missing indicator I_{missing}, Missing set $\mathcal{M}$

Phase 1: Semantic Parsing
$\mathcal{G} \leftarrow \text{ExtractGoal}(q)$
$\mathcal{K} \leftarrow \text{ExtractKnown}(q)$
$\mathcal{T} \leftarrow \text{BuildDependencyGraph}(\mathcal{G}, \mathcal{K})$
Phase 2: Prerequisite Enumeration
$\mathcal{P} \leftarrow \emptyset$
Queue $\leftarrow [\mathcal{G}]$
while Queue $\neq \emptyset$ **do**
  $node \leftarrow$ Queue.pop()
  $prereqs \leftarrow \mathcal{M}(\text{"Prerequisites for: "} + node)$
  for $p \in prereqs$ **do**
    if $p \notin \mathcal{P} \cup \mathcal{K}$ **then**
      $\mathcal{P} \leftarrow \mathcal{P} \cup \{p\}$
      Queue.append(p)
    end if
  end for
end while
Phase 3: Availability Verification
$\mathcal{M} \leftarrow \emptyset$
for $p \in \mathcal{P}$ **do**
  $availability \leftarrow \text{CheckAvailability}(p, \mathcal{K}, \mathcal{M})$
  $confidence \leftarrow \text{ComputeConfidence}(availability)$
  if $confidence < \theta$ **then**
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{p\}$
  end if
end for
Phase 4: Completeness Assessment
$I_{\text{missing}} \leftarrow (\mathcal{M} \neq \emptyset)$
return $I_{\text{missing}}, \mathcal{M}$

Complexity Analysis

Time Complexity

The computational complexity of RT-ICA can be analyzed through its core components. Let d be the maximum depth of the prerequisite dependency tree, b be the average branching factor, and c be the cost of a single LLM call. The worst-case time complexity of RT-ICA is:

$$T(n) = O(c \cdot b^d) \tag{4}$$

where n represents the problem size.

Space Complexity

The space complexity of RT-ICA is dominated by the storage of prerequisites and the dependency graph:

$$S(n) = O(b^d + |\mathcal{K}| + |\mathcal{M}|) \tag{5}$$

where $|\mathcal{K}|$ is the size of known information and $|\mathcal{M}|$ is the size of missing information.

Performance Bounds

We establish formal guarantees for the RT-ICA framework under certain assumptions.

Completeness and Soundness

Definition 1 (ϵ-Completeness). RT-ICA is ϵ-complete if for any problem q with missing information $\mathcal{M}^*$, the algorithm identifies at least $(1-\epsilon)|\mathcal{M}^*|$ missing elements.

Definition 2 (δ-Soundness). RT-ICA is δ-sound if for any problem q, the probability of false positives (incorrectly flagging complete problems as incomplete) is bounded by δ.

Theorem 1 (Performance Bounds). Under the assumption that the language model $\mathcal{M}$ has accuracy α in prerequisite identification and β in availability verification, RT-ICA achieves:

$$\mathbb{E}[|\mathcal{M} \cap \mathcal{M}^*|] \geq \alpha\beta|\mathcal{M}^*| \tag{6}$$

$$\mathbb{P}(\text{false positive}) \leq (1-\alpha)(1-\beta) \tag{7}$$

Implementation Specifications

Goal Extraction Function:

```
Function ExtractGoal(q):
    Parse q to identify question
    phrase (e.g., "how old", "what
    is")
    Extract target variable and
    required computation
    Return formal goal
    specification G
End Function
```

Dependency Graph Construction:

```
Function BuildDependencyGrap-
-h(G, K):
    Initialize graph T with root
    node G
    For each k in K:
```

```
        Add edge k → G if direct
        dependency exists
        Recursively expand
        dependencies using
        template matching
    Return dependency graph T
End Function
```

Availability Verification:

```
Function CheckAvailability(p, K,
M):
    If p in K then return Explic-
    -itlyProvided
    Else if exists derivation
    path from K to p then return
    Derivable
    Else if p in M then return
    AlreadyMissing
    Else return PotentiallyMissing
End Function
```

Confidence Scoring

The confidence score for availability verification combines multiple evidence sources:

$$\text{Confidence}(p) = \lambda_1 S_{\text{sem}} + \lambda_2 S_{\text{syn}} + \lambda_3 S_{\text{con}} \quad (8)$$

where S_{sem} measures the semantic similarity between condition p and the known information $\mathcal{K}$, S_{syn} represents a pattern matching score computed against predefined syntactic templates, and S_{con} evaluates consistency with the broader problem context and domain knowledge. The weighting coefficients λ_i are non-negative and sum to one, i.e., $\sum_{i=1}^{3} \lambda_i = 1, \lambda_i \geq 0$.

Threshold Optimization

The detection threshold θ is optimized to balance precision and recall:

$$\theta^* = \arg\min_{\theta} \left[(1-\alpha)\text{FP}(\theta) + \alpha \text{FN}(\theta)\right] \quad (9)$$

where $\text{FP}(\theta)$ and $\text{FN}(\theta)$ represent false positive and false negative rates at threshold θ, and α controls the trade-off based on application requirements.

B Appendix B: Prompt Templates and Implementation Details

Core Prompt Structure

The RT-ICA framework employs structured prompts that guide the model through systematic reverse thinking:

Phase 1: Problem Understanding

```
Analyze the following problem and
identify:
1. The ultimate question being
asked
2. What a complete solution would
require

Problem: [PROBLEM_TEXT]
```

Phase 2: Prerequisite Enumeration

```
Working backwards from the solution,
list all conditions
that must be true to answer the
question. For each condition,
specify what information would
be needed.

Required conditions:
1. [CONDITION_1] - Requires:
[INFORMATION_NEEDED_1]
2. [CONDITION_2] - Requires:
[INFORMATION_NEEDED_2]
...
```

Phase 3: Availability Verification

```
For each condition above, check
if the required information
is provided in the problem or
can be derived from provided
information. Mark as:
- AVAILABLE: [Condition] -
[Source in problem]
- DERIVABLE: [Condition] -
[Derivation path]
- MISSING: [Condition] -
[What specifically is missing]
```

Model-Specific Adaptations

For different language models, we applied the following adaptations while maintaining the core reasoning structure:

GPT-3.5-turbo: Used more explicit formatting instructions and step-by-step guidance.

GPT-5.1: Leveraged the model's stronger reasoning capabilities with fewer explicit formatting constraints.

DeepSeek-V3.2-Exp: Incorporated the model's native reasoning structure tokens and followed its preferred response format.

Computational Cost Analysis

Detailed computational cost measurements across different model sizes and problem complexities:

Model	Tokens	Time (s)	Accuracy
GPT-3.5-turbo	487	3.2	+27.62%
GPT-5.1	512	4.1	+14.91%
DeepSeek-V3.2-Exp	465	2.8	+13.86%

The token counts represent the total input+output tokens for the complete RT-ICA reasoning process. Processing times are measured end-to-end including API overhead where applicable.

C Appendix C: Ethical Statement and Data Documentation

This appendix details the ethical considerations, data provenance, and annotation protocols related to our benchmark dataset and research.

C.1 Potential Risks and Broader Impacts

The primary goal of our work is **diagnostic and constructive**: to identify a critical blind spot in LLM reasoning (the inability to detect underspecified problems) and to provide tools (a benchmark and a method) to address it. We foresee the following impact profile:

- **Positive Impacts**: By enabling models to "know what they don't know," our research contributes to improving the **robustness, reliability, and transparency** of LLMs. This is a crucial step towards safer deployment in real-world applications where incomplete information is common, such as educational tutoring, decision support, and interactive dialogue systems.
- **Potential Risks and Mitigations**:
  1. **Dual-Use Concern**: The capability to detect missing information could, in principle, be used to craft more sophisticated adversarial examples that deliberately exploit this specific weakness. However, our work **exposes and helps mend this vulnerability**, thereby raising the security baseline. We believe the net effect is strongly positive.
  2. **Misapplication of the Benchmark**: The dataset is intended for **research and evaluation purposes** to diagnose and improve model capabilities. It should not be used as a high-stakes test set without proper validation or as a direct source for training data without consideration of its constructed nature.
  3. **Over-reliance on Automated Detection**: Integrating such a detection module without human oversight in critical applications could pose risks if the module itself errs. Our framework is proposed as an **assistive tool** for enhancing human-AI collaboration, not replacing human judgment.

We release our resources under the **Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0) license** to encourage open research while discouraging immediate commercial deployment without further safeguards.

C.2 Data Provenance, Privacy, and Offensive Content

Our datasets, `test_gsm8k` and `test_math`, are **derivative works** based on two publicly available, established academic benchmarks: **GSM8K** and **Math500**.

- **Source Data Characteristics**: The original datasets consist of **synthetic, grade-school level mathematical word problems**. They do not contain any personal identifiable information (PII), biographical details, or real-world personal data. The narratives are purely fictional and constructed for educational purposes.
- **Our Processing and Checks**: Our methodology involved **systematically removing or altering numerical values and relational phrases** to create "incomplete" variants. This process did not introduce any new personal data, sensitive topics, or offensive language. We manually reviewed a significant sample of the generated problems to ensure they maintained the benign, fictional nature of the source material. No content was found to be offensive, discriminatory, or private.
- **Anonymity**: As the source data contains no PII and our modifications are purely structural/logical, **no anonymization procedure was required** beyond what is already inherent in the original datasets.

C.3 Instructions to Annotators / Data Modifiers

The "annotation" in our work refers to the **structured modification process** performed by the authors and trained research assistants to create incomplete problem instances. All modifiers were made aware that the created data would be used to build a public research benchmark for evaluating language models.

C.4 Consent and Intended Use Communication

- **Source Data**: The original datasets (GSM8K, Math500) were published for academic research use under their respective licenses (MIT, etc.). Our use falls squarely within this intended purpose.

- **Our Modifications**: The individuals involved in modifying the data (the research team) are co-authors or acknowledged contributors. They were fully informed that the resulting dataset would be **publicly released as a benchmark for the NLP research community** to evaluate model capabilities in missing information detection. Their participation implies consent for this use.

- **No External Human Subjects**: Our work did not involve interacting with or collecting data from human subjects beyond the aforementioned structured modification task performed by the research team. Therefore, institutional review board (IRB) approval was not sought, as per common guidelines for this type of data curation research.

This study utilized only publicly available, non-personal datasets and involved no interaction with human subjects. The data modification was performed by the research team as part of standard scholarly activity. Therefore, this research was deemed exempt from institutional review board (IRB) approval in accordance with our institution's policies for research involving existing, anonymized data.