HYDRA: Evaluating Logical Thinking Abilities of LLMs through Formulating Hypotheses and Verification Methods

Shin-nosuke Ishikawa
Graduate School of Artificial Intelligence and Science, Rikkyo University
AI technical sector, Mamezo Co., Ltd. shinnosuke-ishikawa@rikkyo.ac.jp

Hirotsugu Ohba
Graduate School of Artificial Intelligence and Science, Rikkyo University

Masato Todo and **Atsushi Yoshino** and **Masahiro Daimon**
AI technical sector, Mamezo Co., Ltd.

Abstract

Logical consistency in the outputs of large language models (LLMs) is essential for extending their applications beyond single-task assistants, including their use in AI agent systems. Although logical reasoning capabilities are often evaluated through tasks such as mathematics and coding, where it is feasible to assess the correctness of outputs in a well-defined manner, real-world problem solving requires inductive and abductive reasoning, with an emphasis on the ability to formulate and verify hypotheses. When evaluating the hypothesis formulation and verification capabilities of LLMs, the major challenge lies in preparing a variety of tasks with appropriate difficulty, avoiding those that are either too easy or unsolvable. We introduce an automated framework called HYpothesis-Driven Reasoning Assessment (HYDRA), focusing on hypothesis formulation and verification for real-world problem solving with logical reasoning. In HYDRA, all components, including situation generation, hypothesis, and evaluation, are fully performed by LLMs. HYDRA's systematically constructed generation and evaluation strategies enable assessment using realistic, non-trivial situations that offer both diversity and clear criteria. We evaluated six state-of-the-art LLMs using HYDRA and found that their success rates ranged from 70.3% to 93.7%, indicating that even the best-performing model was not perfect and still produced rubric-invalid hypothesis–verification pairs. [1]

1 Introduction

The logical reasoning capabilities of large language models (LLMs) are a key aspect of intelligence and are essential for LLMs to be considered true artificial intelligence (AI). Today is often described as the era of AI agents with autonomous decision-making capabilities, and AIs, including LLMs, are expected to take actions such as gathering information at their own discretion (Wiesinger et al., 2025). In this context, it is crucial for AIs to understand a situation and select appropriate actions for real-world applications. If an AI fails to organize the information it possesses and takes inappropriate actions, its ability to perform a wide range of tasks, including accessing external systems, becomes meaningless—or worse, it may cause harm greater than doing nothing at all.

Various aspects of the logical reasoning abilities of LLMs have been evaluated in previous studies, particularly in tasks related to mathematics (e.g., Cobbe et al., 2021; Li et al., 2024b; Jahin et al., 2025) and coding (e.g., Chen et al., 2021; Jimenez et al., 2024; Yu et al., 2025). In such tasks, all information required to derive correct answers is typically provided, and the correctness of the outputs can be assessed independently of the LLMs themselves. In contrast, real-world problem solving does not guarantee access to all the information needed to derive a unique solution, and decisions often must be made when some of the relevant information is missing. In such cases, reasoning only from the given facts is not sufficient; one must formulate plausible hypotheses about what is not yet known and devise ways to obtain additional information that can verify them. This process relies on inductive and abductive reasoning, and the ability to formulate hypotheses and design verification methods is therefore essential. Having a hypothesis allows an AI agent to take the next action aimed at testing it, and by iteratively formulating and verifying hypotheses based on previous results, the agent can arrive at valuable insights.

Many benchmarks have also been proposed for various domain-specific tasks (Ni et al., 2025). These are valuable for evaluating models in applications aligned with each specific domain. However, it is not clear whether capabilities demonstrated in one domain can be generalized to others. For real-

[1] The code (MIT License), prompts, and generated situations (CC BY 4.0) are available at https://github.com/hidarikawa-p/hydra-bench.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 4090–4105
November 6-10, 2026 ©2026 Association for Computational Linguistics

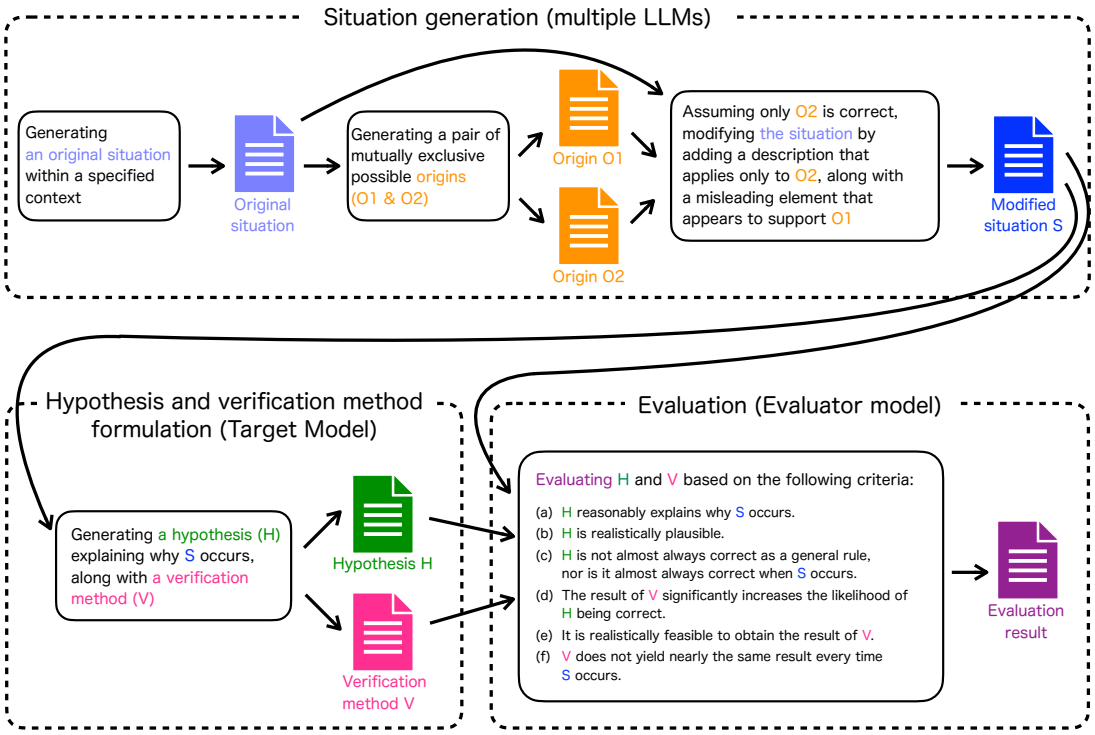


Figure 1: Overview of the autonomous hypothesis and verification method evaluation framework, HYDRA. LLMs are applied in all phases, including situation generation, hypothesis and verification method formulation, and evaluation.

world tasks, where unexpected situations may arise, evaluating general domain-agnostic reasoning abilities provides a complementary axis of assessment to domain-specific benchmarks (Miller and Tang, 2025).

To evaluate logical reasoning abilities in realistic and domain-agnostic settings, it is necessary to prepare a wide variety of situations. If generality alone were the only goal, then abstract representations of tasks, such as descriptions using symbols, would be sufficient. However, in order to assess real-world problem-solving abilities, it is also essential to work with descriptions of realistic situations written in natural language (e.g., Koda et al., 2025). This approach is commonly used in human assessments of logical thinking as well, since demonstrating symbolic reasoning by itself does not guarantee the ability to apply logical thinking in practical contexts. While several studies have assessed the deductive aspects of LLMs' logical reasoning abilities using a wide variety of realistic situation descriptions (Liu et al., 2020, 2023; Yu et al., 2020), applying this approach to the task of hypothesis formulation and verification remains a challenge.

In this paper, we present an automated framework for evaluating hypothesis formulation and verification methods in large language models (LLMs), called HYpothesis-Driven Reasoning Assessment (HYDRA), as summarized in Fig. 1. The HYDRA framework addresses the challenge of assessing logical reasoning capabilities in a wide variety of realistic situations by making full use of LLMs. We also propose an interpretable evaluation strategy that enables detailed analysis of these capabilities. Using the HYDRA framework, we evaluated OpenAI GPT-5.5 Pro (OpenAI, 2026b) and GPT-5.5 (OpenAI, 2026b), Google Gemini 3.1 Pro (Gemini Team, 2023; Deepmind, 2026), Claude Opus 4.7 (Anthropic, 2026), Qwen3 235B A22B 2507 (Yang et al., 2025), DeepSeek-R1 (DeepSeek-AI et al., 2025; Guo et al., 2025), and we present the results in this paper. We note that our evaluation focuses on the validity of the verification design rather than on the execution of the verification itself; the target LLMs are not required to carry out the proposed verification procedures.

2 Related Work

2.1 Inductive and Abductive Reasoning

Several studies have evaluated the logical reasoning abilities of LLMs, with a focus on inductive and

abductive aspects (e.g., Salimi et al., 2026). Abductive NLI (Bhagavatula et al., 2020) is a multiple-choice dataset designed to test the ability to select plausible explanations connecting two observations. LogiEval (Liu et al., 2025a), a comprehensive benchmark covering deductive, inductive, and abductive reasoning capabilities, is also based on a multiple-choice format. Although the multiple-choice format has the advantage of providing rigid ground truth, the HYDRA framework presented in this paper relies on free-form generation to assess genuine hypothesis formulation capabilities.

In addition to multiple-choice style evaluation benchmarks, several evaluation methods have been proposed for assessing inductive and abductive reasoning capabilities without presenting predefined answer options. GEAR (He et al., 2025a) targets abstract symbolic reasoning tasks such as ARC (Chollet, 2019) and ACRE (Zhang et al., 2021), while InductionBench (Hua et al., 2025) and MIRAGE (Li et al., 2024a) also focus on symbolic and abstracted induction and abduction tasks. Another line of research evaluates inductive reasoning through tasks involving the discovery of mathematical regularities (Ishikawa et al., 2025). The InAbHyD (Sun and Saparov, 2025) dataset consists of short sentences composed of simple subject–predicate relationships. We focus on capabilities relevant to real-world tasks and design the benchmark to use realistic situation descriptions rather than symbolic representations.

2.2 Hypothesis Generation for Real-World Tasks

Several studies on hypothesis generation in real-world contexts have also been proposed. The primary concept of HypoBench (Liu et al., 2025b) is closely related to our goal of targeting real-world hypothesis generation; however, the task in HypoBench focuses on deriving insights from structured data. In contrast, the task presented in this paper involves generating hypotheses based on situations described in natural language.

Other studies on domain-specific hypothesis generation have been conducted, such as TruthHypo (Xiong et al., 2025) and SHG (O'Neill et al., 2025) for scientific hypothesis generation, as well as those focusing on the reasoning processes of LLMs through mind tracing (Kim et al., 2025). As shown in the previous section, our goal is to establish a benchmark for general real-world hypothesis generation tasks without specifying a particular domain.

3 HYDRA Framework

To evaluate the capabilities of LLMs to formulate hypotheses and verify them, it is necessary to prepare a situation description that serves as the target condition for hypothesis generation, along with a method for evaluating the LLMs' outputs. Here, we present the proposed HYDRA framework, which includes procedures for preparing situations and evaluating the resulting hypotheses and verification methods. Our goal is to assess abilities related to problem solving, and we do not adopt a multiple-choice format, as real-world problem solving does not provide predefined options to choose from. In the HYDRA framework, LLMs are asked to formulate a hypothesis and a verification method based on a given situation, and their free-form outputs are then evaluated. The complete set of prompts used in the HYDRA framework is provided in Appendix A.

3.1 Situation Generation Method

To function as a benchmark, situation descriptions must be carefully tuned so that realistic hypotheses can be generated, while ensuring that the task is not too easy. In addition, a sufficient number of situations must be prepared to reduce the influence of randomness and uncertainty. It is also important that the situations cover a variety of topics to avoid domain-specific bias and to maintain the generality of the benchmark for evaluating real-world logical thinking abilities. Although a fixed set of situations could in principle be curated manually, we adopt automated generation by LLMs for the following reasons. First, automated generation allows new situation sets to be produced on demand, which mitigates the risk of data contamination once a fixed benchmark becomes publicly available and is potentially included in training data. Second, if models approach saturation, the generation procedure can be rerun or adjusted to produce more challenging situations, which helps extend the lifetime of the benchmark. Third, the set of target domains can be narrowed or expanded according to users' needs, as described below, without additional manual effort. For these reasons, the role of situation preparation is assigned to automated generation by LLMs in the HYDRA framework.

Since hypotheses are typically unnecessary for ordinary situations, we prompted LLMs to generate situations that slightly deviate from the usual. However, we observed that simply asking LLMs to

generate unusual situations often results in outputs with almost no constraints on possible hypotheses, making it too easy to propose one. To address this, we introduced control by preparing a "booby trap": a likely but ultimately implausible origin of the situation (origin O1), along with a plausible and solvable origin (origin O2), ensuring that the problem remains solvable. In HYDRA's situation generation process, the LLM first generates an original situation, then produces two mutually exclusive possible origins, O1 and O2. The LLM then modifies the original situation description so that it is consistent with O2 rather than O1, while including a misleading detail that superficially appears to support O1. We note that the framework is not designed to have LLMs choose between O1 and O2, and these origins are not presented to the target LLMs. Furthermore, although O2 is designated as the correct origin during generation, the evaluation does not require the target answer to match O2, and neither O1 nor O2 is referenced when evaluating the generated hypotheses. An example set consisting of the original situation and the origins O1 and O2 is shown in Fig. 2.

To ensure variety in situation settings, we observed that the generated situations tend to be quite similar, regardless of how many times they are generated. To address this lack of diversity, we specified context keywords in the input prompts when instructing the LLMs to generate situations. To maintain generality, we preselected a set of context keywords based on the Dewey Decimal Classification (Dewey, 1876; OCLC, 2011), a widely used library classification system that categorizes human knowledge. We manually selected five keywords from each of the top-level classes to ensure topic diversity, excluding the religion class. We excluded religion because it deals with faith and belief, which we considered outside the scope of logical reasoning and better addressed separately. We also avoided keywords that specify particular regions or cultures. As a result, we obtained 45 context keywords, as shown in Table 1. We note that users are not required to use the 45 context keywords in every evaluation. If users wish to focus on a specific domain or a set of related fields of interest, they have the option to narrow down or expand the set of context keywords accordingly.

Additionally, we used multiple LLMs to generate the set of situations. We confirmed that the outputs differ across models even when using the same context, which helped ensure both the quantity and the variety of the generated situations.

3.2 Evaluation Method

To evaluate the free-form outputs of simulated real-world tasks, it is not feasible to apply rule-based methods. By analogy with human free-form assessments, we opted to have an evaluator assess the hypotheses and verification methods based on the given situations and evaluation criteria. Considering availability and scalability, the HYDRA framework employs an LLM-as-a-judge as the evaluator (Zheng et al., 2023). While this choice trades off perfect objectivity, it enables scalable evaluation across diverse free-form outputs. It has been noted that LLM-as-a-judge strategies work well for a variety of tasks, and that prompt design can significantly improve the quality of LLM-based evaluations (Li et al., 2025). Below, we present the set of evaluation criteria included in the prompt for assessing the formulated hypotheses and their corresponding verification methods.

These criteria are defined based on Bayes' theorem. Given a situation S (corresponding to the modified situation described in the previous subsection) and a hypothesis H, we require that the conditional probability $P(H|S)$, the probability that H is correct given that S occurs, is reasonably high. According to Bayes' theorem, we obtain the following:

$$P(H|S) = \frac{P(S|H)P(H)}{P(S)}. \quad (1)$$

For $P(H|S)$ to be high, both $P(S|H)$ and $P(H)$ must be reasonably high, relative to the scale set by $P(S)$. Corresponding to these factors, we define the following criteria:

(a) The hypothesis reasonably explains why the situation occurs.

(b) The hypothesis is realistically plausible.

We do not consider $P(S)$ since it depends only on S, and we assume that S has occurred.

Furthermore, when $P(H)$ or $P(H|S)$ is close to 1, the hypothesis could be trivial. For example, a hypothesis stating that a man is not married, given a situation where the man is single, yields a certainty that makes the hypothesis uninformative. To avoid such triviality, we add the following criterion:

(c) The hypothesis is not almost always correct as a general rule, nor is it almost always correct when the situation occurs.

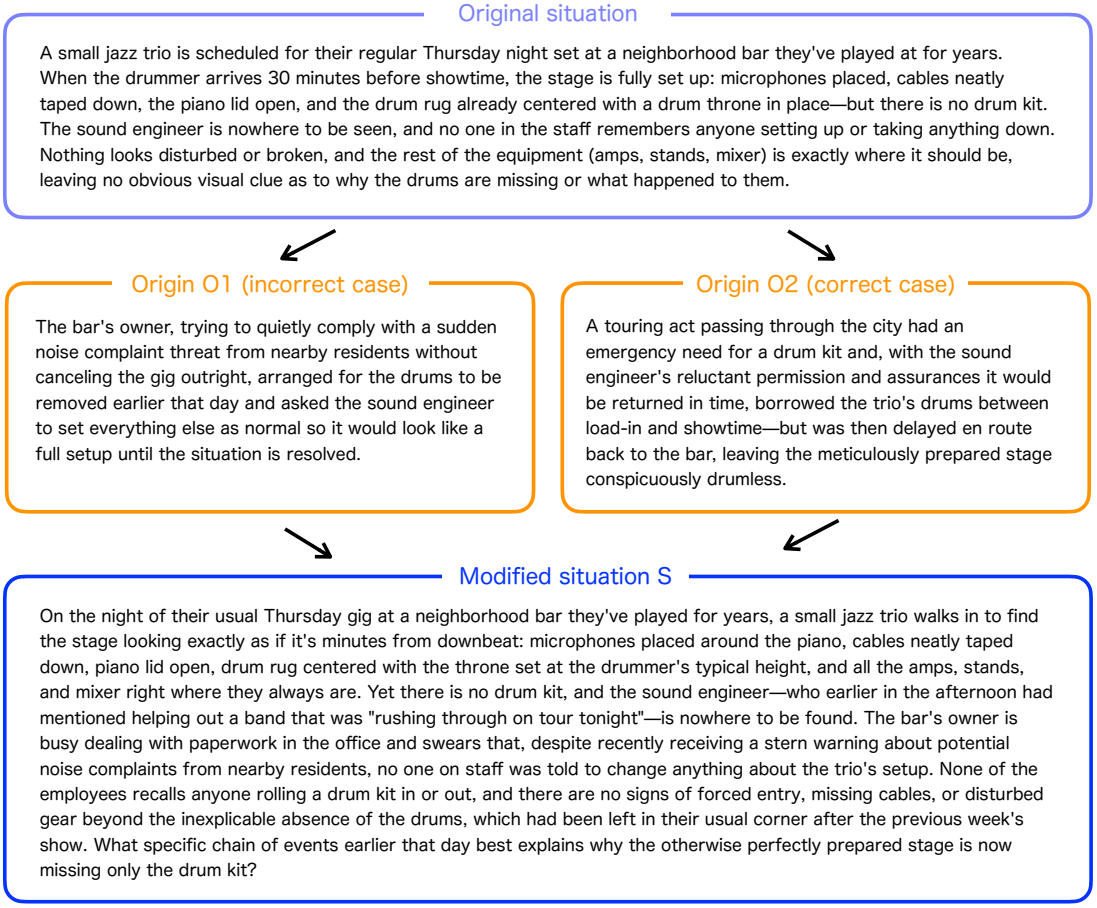


Figure 2: Example of the situation generation process, including intermediate steps, taken from the experiment described in Section 4. The situation generator LLM was GPT-5.1, and the input context keyword was "music." Note that origins O1 and O2 are not shown to the target LLM.

000 Computer science, information and general works	**400 Language**	**700 Arts and recreation**
1. Computer programming	16. Language	31. Architecture
2. Libraries	17. Foreign languages	32. Painting
3. Artificial intelligence	18. Classical languages	33. Photography
4. Magazines	19. Dictionaries	34. Music
5. News media	20. Grammar	35. Sports
100 Philosophy and psychology	**500 Pure science**	**800 Literature**
6. Philosophy	21. Natural science	36. Classical literature
7. Psychology	22. Mathematics	37. Poetry
8. Dreams	23. Physics	38. Science fiction
9. Logic	24. Chemistry	39. Novels
10. Ethics	25. Biology	40. Essays
300 Social sciences	**600 Technology**	**900 History and geography**
11. The environment	26. Health and medicine	41. Historical documents
12. Politics	27. Engineering	42. Geography and travel
13. Economics	28. Farming and agriculture	43. Ancient history
14. Education	29. Management	44. Biographies
15. Customs and etiquette	30. Manufacturing	45. Modern history

Table 1: List of the 45 manually selected context keywords used to ensure variation in situation generation. The nine categories are based on the Dewey Decimal Classification, with five keywords selected for each category.

For the verification method, we similarly define the evaluation criteria based on Bayes' theorem. Given a successful result of the verification method V, we require that the probability of the hypothesis

being true, $P(H|S,V)$, is sufficiently high. Note that we do not require the verification method to be perfect; specifically, $P(H|S,V)$ does not need to be 1. Applying Bayes' theorem, we have:

$$P(H|S,V) = \frac{P(V|S,H)P(H|S)}{P(V|S)} \quad (2)$$

To make $P(H|S,V)$ high, we require that $P(V|S,H)$ is high, relative to the scale set by $P(V|S)$. Based on this, we define the following criterion:

(d) The result of the verification method significantly increases the likelihood of the hypothesis being correct.

In addition, we require that the verification method be realistically executable and that it produces a conclusive outcome (either V or not V):

(e) It is realistically feasible to obtain the result of the verification method.

Finally, similar to criterion (c), we require that $P(V|S)$ is not always true in order to avoid uninformative verification:

(f) The verification method does not yield nearly the same result every time the situation occurs.

We include the six criteria (a)–(f) in the input prompt for the evaluator LLM and ask it to output an evaluation result for each of them. We then determine the final evaluation result by checking whether all six criteria are marked as OK. If any criterion is marked as NG, the hypothesis formulation and verification process is considered to have failed.

4 Experiments

4.1 Experimental Settings

We present an experiment to evaluate the logical reasoning abilities of recent LLMs through the formulation of hypotheses and verification methods using the HYDRA framework. We selected six state-of-the-art LLMs for this experiment to evaluate their capabilities in formulating hypotheses and verification methods: OpenAI GPT-5.5 Pro, GPT-5.5, Google Gemini 3.1 Pro, Anthropic Claude Opus 4.7, Qwen3 235B A22B 2507, and DeepSeek-R1. For situation generation, we used a dataset prepared in advance before some of the above models were released. The dataset was generated by OpenAI GPT-5.2 Pro (OpenAI, 2025a,c), GPT-5.1 (OpenAI, 2025a,b), Google Gemini 3 Pro (Deepmind, 2025), Anthropic Claude Opus 4.5 (Anthropic, 2025), Qwen3 235B A22B 2507, and DeepSeek-R1. We used this dataset because the situations served only as target topics for evaluation, and using the latest models merely for marginal improvements in quality was not essential. All models were accessed via API: OpenAI, Google, and Anthropic models through their official APIs, and the other models via Amazon Bedrock. All configurable parameters were set to their default values except for the maximum output length of the Anthropic models (16,384 tokens), which must be specified.

We constructed a list of situations for the experiment using the six models, covering all the contexts shown in Table 1 by generating one situation per LLM for each context (i.e., one situation per (context, generator) pair). One instance of situation generation by Claude Opus 4.5 for the biology context failed due to a content filter, resulting in a total of 269 situations (45 contexts $\times$ 6 models $-$ 1 failure). Some outputs of the situation generator contained a title or label before the requested single paragraph, or additional material after it, such as answer options or statements of the intended origins. We retained only the situation paragraph and removed such material before presenting the situations to the target models.

Using this dataset, we prompted each of the six LLMs to generate a hypothesis and a verification method for each situation using the following simple instruction: "Provide the single most plausible hypothesis for why the following situation occurs, along with a method of verification." Evaluation was conducted using the method described in the previous section. We selected Claude Opus 4.7 as the evaluator because it achieved the highest final judgment accuracy among the evaluated models in a preliminary experiment using manually prepared hypothesis–verification method pairs. Details of the preliminary experiment are provided in Appendix B.

To examine the effect of evaluator selection on the results, including the effect of self-evaluation, we performed the evaluation using another evaluator model. Although we found that models tended to evaluate themselves more favorably than the other evaluators did, the score ranking remained the same when the evaluators themselves were excluded. Details are provided in Appendix C.

4.2 Results

We successfully evaluated the hypothesis formulation and verification capabilities of LLMs using the HYDRA framework and obtained the success rates for evaluation criteria (a)–(f), as well as the final evaluation results. Figure 3 presents examples of both successful and unsuccessful cases of formulated hypothesis and verification method pairs. According to the evaluator, the successful pair includes a plausible hypothesis together with a reasonable verification method, whereas the unsuccessful pair is evaluated as having a hypothesis judged as (b) unrealistic and a verification method judged as (e) difficult to perform realistically.

The results are summarized in Table 2. All LLMs successfully generated pairs for all 269 situations, and all of them were successfully evaluated. In total, we obtained 1,614 evaluated cases. The tested LLMs demonstrated high performance, with average success rates above 90% for all six criteria. Notably, all models achieved perfect or nearly perfect scores on criterion (c), indicating that they did not produce trivial or uninformative hypotheses in most cases.

In contrast, the average success rate for the final result remained relatively low at 84.9%. This gap suggests that in many failure cases, only one criterion was not satisfied. For example, a hypothesis might explain the situation but lack realism (criterion (b) failure), or the verification method might be impractical despite a reasonable hypothesis (criterion (e) failure). Such single-criterion failures accounted for 13.6% of all cases, while failures involving multiple criteria occurred in only 1.5%. These results highlight the inherent difficulty of the hypothesis and verification method formulation.

Among the evaluated models, GPT-5.5 Pro achieved the highest final success rate, at 93.7%. GPT-5.5 Pro has been reported to perform well on several benchmarks, and it also demonstrated strong performance in this task. It is worth noting, however, that GPT-5.5 Pro is significantly more expensive than the other models, costing six times more than the second most expensive model, GPT-5.5. In addition, the response speed of GPT-5.5 Pro was significantly slower than that of the other evaluated models, which limits its use cases. For real-world applications, balancing performance, cost, and response speed is an important consideration when selecting an LLM. In contrast, Qwen3 235B A22B 2507 recorded the lowest final success rate among the models tested, at 70.3%. The main reason for this low score was its poor performance on the realistic hypothesis and verification criteria, namely criteria (b) and (e).

Although we confirmed the strong performance of the latest large models, the HYDRA framework is not limited to evaluating large models. We provide examples of evaluations of smaller models in Appendix D.

5 Discussion

As shown in the previous sections, we successfully formulated a method for evaluating the formulation of hypotheses and verification methods. The fact that we did not obtain 100% success indicates two things: first, that the HYDRA framework functions effectively by ensuring that not all situations lead to trivial hypotheses; and second, that even the best-performing model among those evaluated occasionally fails to produce a valid hypothesis–verification pair.

Another feature of HYDRA is that, because the final judgment is decomposed into six criteria, it indicates which requirement a failed hypothesis–verification pair did not satisfy, such as the realism of the hypothesis or the feasibility of the verification method. As shown in Section 4.2, most failures are attributable to a single criterion, with criteria (e) and (b) being the most common sources of failure. Although this criterion-level diagnosis does not identify where the underlying reasoning process breaks down, it provides a finer-grained view of model behavior than a single overall score and may suggest directions for future improvements.

The establishment of an evaluation method for general hypothesis formulation and verification in real-world situations, as presented in this paper, could contribute to broader applications of AI agents. Several studies have pointed out that logical reasoning capabilities, including hypothesis formulation, enhance the performance of AI agents (He et al., 2025b; He and Chen, 2025; Abdaljalil et al., 2025). We expect that well-designed hypothesis generation tasks and their evaluation will further advance the applicability of AI agents in real-world scenarios.

We calculated the average success rates for each context category across the six LLMs tested in the experiment, as summarized in Table 3. We observe that the success rates vary by category, with higher scores in Technology, Pure Science and Computer

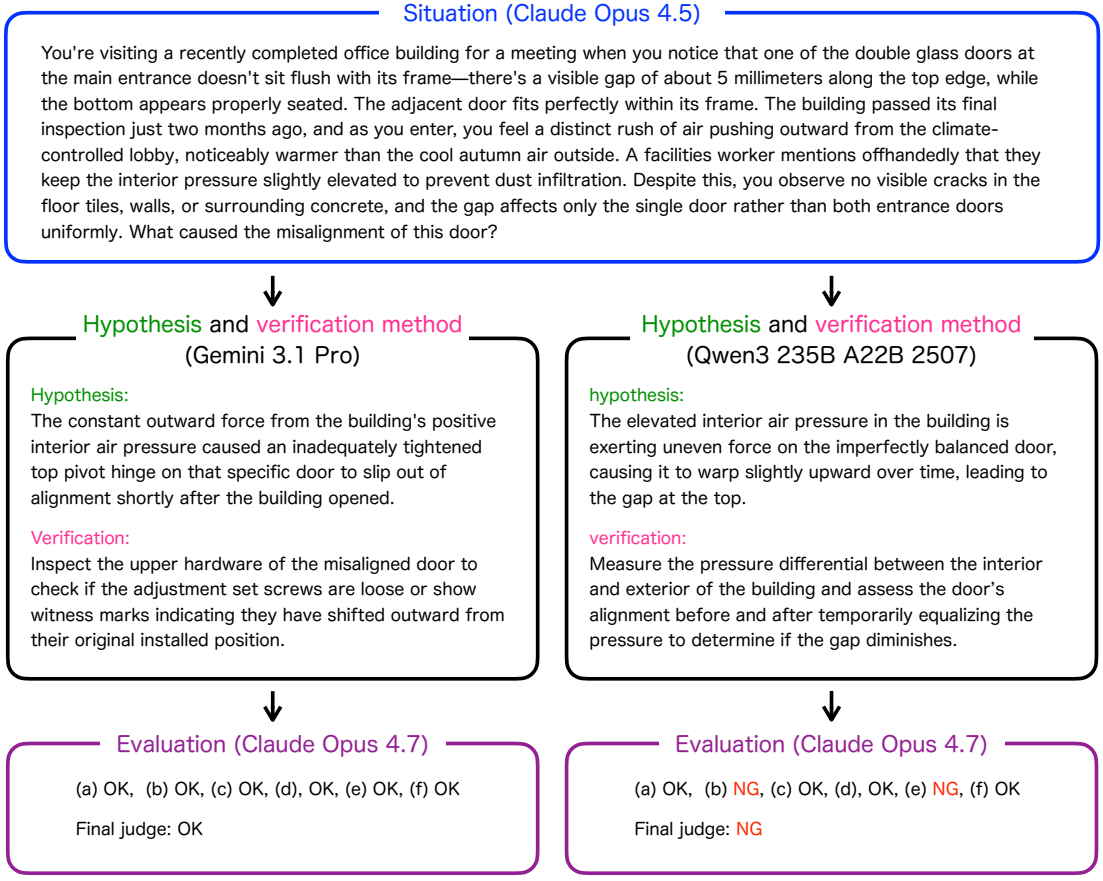


Figure 3: Examples of successful and unsuccessful cases of hypothesis and verification method formulation, along with evaluation results. The situation generator LLM was Claude Opus 4.5 and the input context keyword was "architecture," the target LLMs were Gemini 3.1 Pro and Qwen3 235B A22B 2507, and the evaluator LLM was Claude Opus 4.7.

Model	**(a)**	**(b)**	**(c)**	**(d)**	**(e)**	**(f)**	**Final judge**
GPT-5.5	**100%**	98.1%	99.6%	**100%**	94.4%	**100%**	92.9%
GPT-5.5 Pro	**100%**	**98.9%**	**100%**	**100%**	94.4%	**100%**	**93.7%**
Gemini 3.1 Pro	**100%**	97.0%	**100%**	95.2%	**94.8%**	99.6%	88.1%
Claude Opus 4.7	**100%**	97.8%	**100%**	98.9%	92.9%	99.6%	90.0%
Qwen3 235B A22B 2507	97.8%	92.9%	**100%**	94.4%	80.7%	**100%**	70.3%
DeepSeek-R1	99.3%	93.7%	**100%**	96.3%	82.9%	**100%**	74.3%
Average	99.5%	96.4%	99.9%	97.5%	90.0%	99.9%	84.9%

Table 2: Success rates for hypothesis and verification method formulation across the six tested LLMs. Results for evaluation criteria (a) through (f) are shown separately, along with the final evaluation results. Bold numbers indicate the best score for each criterion and the final result.

Science, and the lowest score in the Literature category. This variation may reflect the extent of data diversity used during training for each category. LLMs are typically well-trained on Technology and Computer Science content through abundant online resources. In contrast, one possible explanation for the lower Literature score is that training data for the Literature category may be less diverse and more limited in quantity.

We note that the goal of this work is not to maximize the performance of hypothesis and verification method formulation, but to establish a framework for evaluating these capabilities with real-world problem solving in mind. Exploring ways to enhance these capabilities using the HYDRA method is one of the directions for future work.

Category	Final judge
000 Computer science, information and general works	91.1%
100 Philosophy and psychology	81.1%
300 Social sciences	78.3%
400 Language	86.7%
500 Pure science	92.0%
600 Technology	93.9%
700 Arts and recreation	86.7%
800 Literature	75.6%
900 History and geography	78.9%

Table 3: Final success rates averaged across the six tested LLMs, categorized by the input context keywords.

6 Conclusion

We constructed a framework called HYDRA to evaluate hypothesis formulation and verification methods, with an emphasis on practical applicability to real-world tasks. By utilizing LLMs for both situation generation and evaluation through systematically designed procedures, we successfully addressed the challenges of maintaining a variety of realistic task situations and establishing clearly defined evaluation criteria. Among the tested LLMs, we found that current models do not exhibit perfect capability when prompted directly, without additional guidance. The HYDRA framework enables scalable, criteria-based evaluation and is expected to serve as a valuable evaluation axis, particularly for AI agent applications.

Limitations

In this paper, we presented a framework for evaluating the capabilities of hypothesis formulation and verification method design. Within this framework, we used LLMs to generate situations and perform evaluations. However, since it is not guaranteed that LLMs can execute these tasks perfectly, there is a non-negligible probability that some situation descriptions may be invalid or that evaluation judgments may be inaccurate. Such imperfections introduce uncertainty into the assessment of LLM capabilities.

As shown in Appendix C, LLM-as-a-judge evaluations exhibit model dependency, including self-preference. Therefore, the resulting scores should be interpreted as estimates from the perspective of the evaluator model.

The framework evaluates single-step reasoning. Multi-step reasoning processes, such as executing verification procedures and revising hypotheses based on the results, are outside the scope of this paper. These aspects are also important and should be investigated in future work.

The situations generated in our experiments are not uniformly diverse. Although context keywords were used to promote variety, a noticeable share of the situations (roughly one in six) followed a similar pattern, in which a facility was unexpectedly closed during its usual opening hours. This pattern resembles one of the examples given in the situation generation prompt, suggesting that the examples in the prompt may bias the generated situations. More systematic approaches to increasing diversity, such as varying or removing the examples or filtering generated situations by semantic similarity, remain to be explored.

We did not evaluate human performance in hypothesis formulation or verification-method design, which limits the interpretation of the model evaluation results.

The capabilities evaluated by this framework include commonsense recognition and assessments of plausibility, rather than purely logical reasoning. This design is intentional, as the framework aims to evaluate real-world problem-solving capabilities.

While the framework targets hypothesis and verification capabilities, it does not comprehensively assess other aspects of logical reasoning, such as causal inference, long-term planning, or consistency maintenance.

Additionally, our experiments were conducted using a selected set of LLMs, and we cannot conclude that the best-performing model in our study is the best among all existing LLMs. Moreover, variations in prompt design for hypothesis and verification method generation, or other optimization strategies, could significantly affect the evaluation results.

Ethics Statement

In this study, we aimed to evaluate logical reasoning abilities and did not use any input containing personal information, aside from the context keywords listed in Table 1. We did not intend for the LLMs to engage in ethically sensitive tasks, and we have identified no specific ethical concerns related to the methods presented.

Acknowledgements

We thank Masao Yasui for valuable discussions on the application of LLMs, including AI agents, to real-world tasks. We are also grateful to Taiki Ogihara for suggesting that using a multiple-choice format does not truly evaluate hypothesis formulation. This work was supported by JSPS Kakenhi, Grant Number 24K15077. This research was inspired by *The BCG Way: The Art of Hypothesis-Driven Management* by Kazunari Uchida. We used ChatGPT and Claude as assistants for grammar checking and text editing, and the released code was generated with the assistance of Claude.

References

Samir Abdaljalil, Hasan Kurban, Khalid Qaraqe, and Erchin Serpedin. 2025. Theorem-of-thought: A multi-agent framework for abductive, deductive, and inductive reasoning in language models. In *Proceedings of the 3rd Workshop on Towards Knowledgeable Foundation Models (KnowFM)*, pages 111–119, Vienna, Austria. Association for Computational Linguistics.

Anthropic. 2025. System Card: Claude Opus 4.5. https://assets.anthropic.com/m/64823ba7485345a7/Claude-Opus-4-5-System-Card.pdf Accessed: Jan 5, 2026.

Anthropic. 2026. System Card: Claude Opus 4.7. https://anthropic.com/claude-opus-4-7-system-card Accessed: May 25, 2026.

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*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code.

François Chollet. 2019. On the measure of intelligence. *Preprint*, arXiv:1911.01547.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Google Deepmind. 2025. Gemini 3 Pro Model Card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf Accessed: Jan 5, 2026.

Google Deepmind. 2026. Gemini 3.1 Pro Model Card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-1-Pro-Model-Card.pdf Accessed: May 25, 2026.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *Preprint*, arXiv:2501.12948.

Melvil Dewey. 1876. *A Classification and Subject Index for Cataloguing and Arranging the Books and Pamphlets of a Library*. s.n., Amherst, Mass. First edition of the Dewey Decimal Classification.

Gemini Team. 2023. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638.

Kaiyu He and Zhiyu Chen. 2025. From reasoning to learning: A survey on hypothesis discovery and rule learning with large language models. *Preprint*, arXiv:2505.21935.

Kaiyu He, Peilin Wu, Mian Zhang, Kun Wan, Wentian Zhao, Xinya Du, and Zhiyu Chen. 2025a. Gear: A general evaluation framework for abductive reasoning. *Preprint*, arXiv:2509.24096.

Kaiyu He, Mian Zhang, Shuo Yan, Peilin Wu, and Zhiyu Zoey Chen. 2025b. IDEA: Enhancing the Rule Learning Ability of Large Language Model Agent through Induction, Deduction, and Abduction. *Preprint*, arXiv:2408.10455.

Wenyue Hua, Tyler Wong, Fei Sun, Liangming Pan, Adam Jardine, and William Yang Wang. 2025. InductionBench: LLMs fail in the simplest complexity

class. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26526–26546, Vienna, Austria. Association for Computational Linguistics.

Shin-nosuke Ishikawa, Masato Todo, Taiki Ogihara, and Hirotsugu Ohba. 2025. Idola tribus of AI: Large language models tend to perceive order where none exists. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12714–12727, Suzhou, China. Association for Computational Linguistics.

Afrar Jahin, Arif Hassan Zidan, Wei Zhang, Yu Bao, and Tianming Liu. 2025. Evaluating mathematical reasoning across large language models: A fine-grained approach. *Preprint*, arXiv:2503.10573.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. 2024. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*.

Hyunwoo Kim, Melanie Sclar, Tan Zhi-Xuan, Lance Ying, Sydney Levine, Yang Liu, Joshua B. Tenenbaum, and Yejin Choi. 2025. Hypothesis-driven theory-of-mind reasoning for large language models. In *COLM*.

Miho Koda, Yu Zheng, Ruixian Ma, Mingyang Sun, Devesh Pansare, Fabio Duarte, and Paolo Santi. 2025. LocationReasoner: Evaluating LLMs on Real-World Site Selection Reasoning. *Preprint*, arXiv:2506.13841.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025. From generation to judgment: Opportunities and challenges of LLM-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2757–2791, Suzhou, China. Association for Computational Linguistics.

Jiachun Li, Pengfei Cao, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. 2024a. MIRAGE: evaluating and explaining inductive reasoning process in language models. *CoRR*, abs/2410.09542.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. 2024b. GSM-Plus: A Comprehensive Benchmark for Evaluating the Robustness of LLMs as Mathematical Problem Solvers. *Preprint*, arXiv:2402.19255.

Hanmeng Liu, Yiran Ding, Zhizhang Fu, Chaoli Zhang, Xiaozhang Liu, and Yue Zhang. 2025a. Evaluating the logical reasoning abilities of large reasoning models. *Preprint*, arXiv:2505.11854.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023. LogiQA 2.0—An Improved Dataset for Logical Reasoning in Natural Language Understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962.

Haokun Liu, Sicong Huang, Jingyu Hu, Yangqiaoyu Zhou, and Chenhao Tan. 2025b. Hypobench: Towards systematic and principled benchmarking for hypothesis generation. *Preprint*, arXiv:2504.11524.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. LogiQA: A Challenge Dataset for Machine Reading Comprehension with Logical Reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3622–3628. International Joint Conferences on Artificial Intelligence Organization. Main track.

Justin K Miller and Wenjia Tang. 2025. Evaluating LLM Metrics Through Real-World Capabilities. *Preprint*, arXiv:2505.08253.

Shiwen Ni, Guhong Chen, Shuaimin Li, Xuanang Chen, Siyi Li, Bingli Wang, Qiyao Wang, Xingjian Wang, Yifan Zhang, Liyang Fan, Chengming Li, Ruifeng Xu, Le Sun, and Min Yang. 2025. A survey on large language model benchmarks. *Preprint*, arXiv:2508.15361.

OCLC. 2011. *Dewey Decimal Classification and Relative Index*, 23rd edition. OCLC, Dublin, OH.

Charles O'Neill, Tirthankar Ghosal, Roberta Răileanu, Mike Walmsley, Thang Bui, Kevin Schawinski, and Ioana Ciucă. 2025. Sparks of science: Hypothesis generation using structured paper data. *Preprint*, arXiv:2504.12976.

OpenAI. 2025a. GPT-5 System Card. https://cdn.openai.com/gpt-5-system-card.pdf Accessed: Jan 5, 2026.

OpenAI. 2025b. GPT-5.1 Instant and GPT-5.1 Thinking System Card Addendum. https://cdn.openai.com/pdf/4173ec8d-1229-47db-96de-06d87147e07e/5_1_system_card.pdf Accessed: Jan 5, 2026.

OpenAI. 2025c. Update to GPT-5 System Card: GPT-5.2. https://cdn.openai.com/pdf/3a4153c8-c748-4b71-8e31-aecbde944f8d/oai_5_2_system-card.pdf Accessed: Jan 5, 2026.

OpenAI. 2026a. GPT-5.4 Thinking System Card. https://deploymentsafety.openai.com/gpt-5-4-thinking/gpt-5-4-thinking.pdf Accessed: May 25, 2026.

OpenAI. 2026b. GPT-5.5 System Card. https://deploymentsafety.openai.com/gpt-5-5/gpt-5-5.pdf Accessed: May 25, 2026.

Moein Salimi, Shaygan Adim, Danial Parnian, Nima Alighardashi, Mahdi Jafari Siavoshani, and Mohammad Hossein Rohban. 2026. Wiring the 'Why': A Unified Taxonomy and Survey of Abductive Reasoning in LLMs. *Preprint*, arXiv:2604.08016.

Yunxin Sun and Abulhair Saparov. 2025. Language models do not follow occam's razor: A benchmark for inductive and abductive reasoning. *Preprint*, arXiv:2509.03345.

Julia Wiesinger, Patrick Marlow, and Vladimir Vuskovic. 2025. Agents. https://www.kaggle.com/whitepaper-agents Accessed: Jan 5, 2026.

Guangzhi Xiong, Eric Xie, Corey Williams, Myles Kim, Amir Hassan Shariatmadari, Sikun Guo, Stefan Bekiranov, and Aidong Zhang. 2025. Toward reliable scientific hypothesis generation: Evaluating truthfulness and hallucination in large language models. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 7849–7857. International Joint Conferences on Artificial Intelligence Organization. Main Track.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 Technical Report. *Preprint*, arXiv:2505.09388.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. ReClor: A Reading Comprehension Dataset Requiring Logical Reasoning. *Preprint*, arXiv:2002.04326.

Zhaojian Yu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. 2025. HumanEval pro and MBPP pro: Evaluating large language models on self-invoking code generation task. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13253–13279, Vienna, Austria. Association for Computational Linguistics.

Chi Zhang, Baoxiong Jia, Mark Edmonds, Song-Chun Zhu, and Yixin Zhu. 2021. ACRE: Abstract Causal REasoning Beyond Covariation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Preprint*, arXiv:2306.05685.

A Complete Set of Prompts Used in HYDRA

All prompts used in the HYDRA framework for the experiments described in Section 4 are provided in the following subsections.

A.1 Prompt for Generating the Original Situation

```
Provide an example of a realistic and likely
scenario that slightly deviates from the usual
in the context of {context keyword}.
For instance, you might find the road wet
when leaving your home in the afternoon,
even though the weather was clear in the
morning, or you might find a café closed
during its normal business hours.
Make sure the cause of the situation cannot
be easily determined through simple
observation, such as by looking around.
Also, avoid subjective examples involving
personal feelings or thoughts; choose an
objective and observable situation instead.
Describe the situation in a few sentences,
including some background details.
```

A.2 Prompt for Generating Possible Origins

```
Provide two possible origins of the following
situation. Do not assume that the described
facts in the situation are incorrect, such as
due to a misunderstanding. Keep both origins
realistically plausible and mutually exclusive
in principle. Avoid any origin that, if true,
would clearly reveal itself as the cause.
For example, if the situation is "the train did
not arrive at the usual time without any
announcement," the origin should not be
"the schedule was moved forward only for
today," since such a change would normally
be announced.Describe the result in one
sentence each, using the following format:
origin1:
origin2:

situation: {original situation}
```

A.3 Prompt for Modifying the Situation

```
Create a question that asks for the origin of
a situation by adding to, modifying, or
reorganizing the given description. Ensure that
origin2 is the correct answer and that origin1
is incorrect. At the same time, include a
misleading detail that makes origin1 appear
plausible, so the question is not too easy.
Present the situation as a single paragraph,
without including any instructions or options
such as "Answer the origin of this situation."

situation: {original situation}
origin1: {origin 1}
origin2: {origin 2}
```

A.4 Prompt for Formulating the Hypothesis and Verification Method

```
Provide the single most plausible hypothesis
for why the following situation occurs, along
with a method of verification. Describe each
in a single sentence, using the following
format:
hypothesis:
verification:

{modified situation}
```

A.5 Prompt for Evaluation

```
Evaluate the hypothesis and the method used
to verify it based on the following criteria:
(a) The hypothesis reasonably explains
why the situation occurs.
(b) The hypothesis is realistically plausible.
(c) The hypothesis is not almost always correct
as a general rule, nor is it almost always correct
when the situation occurs.
(d) The result of the verification method
significantly increases the likelihood of the
hypothesis being correct.
(e) It is realistically feasible to obtain the
result of the verification method.
(f) The verification method does not yield
nearly the same result every time the situation
occurs.
Describe each result using only "OK" or "NG,"
following the format below:
a: OK or NG
b: OK or NG
...
f: OK or NG

situation: {modified situation}
hypothesis: {hypothesis}
verification: {verification method}
```

B Preliminary Experiment for Evaluator LLM Selection

To select the best evaluator for the experiments conducted under the LLM-as-a-judge scheme, we performed a preliminary study to measure evaluation accuracy using a human-created and human-annotated dataset. Each entry in the dataset consisted of a situation description accompanied by a hypothesis–verification method pair. We prepared seven such pairs for each situation: one case that satisfied all six evaluation criteria, referred to as the "all-OK" case, and six "NG" cases, each designed to fail at least one of criteria (a) through (f). We prepared 15 situations in various contexts, yielding a total of $15 \times 7 = 105$ data entries. One example of the situations used was: "I found a café closed during its normal business hours. It was Sunday, and the café is usually open on weekends." The seven hypothesis–verification method pairs corresponding to this example situation, which were used to evaluate the models, are shown in Table 4, together with their annotations.

We compared the capabilities of the LLMs as evaluators using three metrics: accuracy on the individual evaluation criteria (a) through (f), resulting in 630 criterion-level evaluations in total (6 criteria $\times$ 105 cases); accuracy on the final evaluation results across the 105 cases; and the excess OK rate calculated solely from the final judgments, without reference to the criterion-level evaluations, as the difference between the number of cases judged as OK and the 15 annotated OK cases, divided by 15. As evaluator candidates, we compared six individual LLMs. For Claude Opus 4.7, which has an optional thinking mode, we tested the model with and without this mode separately.

The results of the evaluator-comparison experiment are shown in Table 5. As shown, Claude Opus 4.7 without thinking mode achieved the highest final evaluation accuracy among the tested settings. We found that the model with the highest accuracy on the individual criteria was not identical to the model with the highest final judgment accuracy. We decided to prioritize final judgment accuracy because the individual criteria were designed to support the final judgment, and the final judgment is more important for our purpose. Although GPT-5.5 also achieved high final judgment accuracy, its excess OK rate was significantly higher than that of Claude Opus 4.7, indicating that using GPT-5.5 as the evaluator would lead to an excessively high estimated success rate. In the HYDRA framework, it is important to correctly judge invalid hypothesis–verification pairs as invalid. We therefore selected Claude Opus 4.7 as the evaluator for the experiments presented in Section 4. To examine the effect of evaluator model selection, we also compared the evaluation results obtained using GPT-5.5 in Appendix C.

C Evaluator Model Dependency Investigation

It is difficult in principle to ensure perfect accuracy in LLM-as-a-judge evaluation, and the HYDRA framework involves uncertainties, especially regarding the choice of evaluator model. To in-

#	Hypothesis	Verification	(a)	(b)	(c)	(d)	(e)	(f)	Final
1	The café owner had an urgent matter and was unable to open the café or provide an explanation.	Ask the owner at the next opportunity whether he had an urgent matter on that day.	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**
2	It was the last Sunday of the month.	Check a smartphone calendar app.	NG	**OK**	**OK**	**OK**	**OK**	**OK**	NG
3	A new law prohibits cafés and restaurants from opening on Sundays.	Check the news to confirm whether such a law has been enacted.	**OK**	NG	**OK**	**OK**	**OK**	NG	NG
4	It was not possible to enter the café.	Try to enter the café and confirm whether it is accessible.	NG	**OK**	NG	**OK**	**OK**	NG	NG
5	The café owner had an urgent matter and was unable to open the café or provide an explanation.	Check whether there are any signs that a notification was posted.	**OK**	**OK**	**OK**	NG	**OK**	**OK**	NG
6	The café owner had an urgent matter and was unable to open the café or provide an explanation.	Review the café owner's activity log to see if there was an urgent matter.	**OK**	**OK**	**OK**	**OK**	NG	**OK**	NG
7	The café owner had an urgent matter and was unable to open the café or provide an explanation.	Confirm that there is no alternative explanation, such as a sudden change in the calendar itself, including redefined weekdays or unexpected holidays.	**OK**	**OK**	**OK**	NG	**OK**	NG	NG

Table 4: Examples of the content and annotations of the human-created and human-annotated hypothesis–verification method pairs for one example situation used in the preliminary experiment for evaluator selection.

Model	(a)–(f) accuracy	Final accuracy	Excess OK rate
GPT-5.5	85.9%	94.3%	+40.0%
GPT-5.5 Pro	87.3%	89.5%	+73.3%
Gemini 3.1 Pro	**87.4%**	87.6%	+73.3%
Claude Opus 4.7	84.4%	**95.2%**	**−6.7%**
Claude Opus 4.7 (thinking mode)	84.0%	89.5%	+20.0%
Qwen3 235B A22B 2507	70.5%	80.9%	+40.0%
DeepSeek-R1	81.1%	82.9%	+80.0%

Table 5: Evaluation results for the six LLMs, including the thinking and non-thinking settings of Claude Opus 4.7, based on three metrics: accuracy on the individual evaluation criteria, final evaluation accuracy, and excess OK rate compared with the annotations. Boldface indicates the best score for each metric.

vestigate this aspect, we performed an additional evaluation experiment using a different evaluator model. As shown in Appendix B, GPT-5.5 was the second-best evaluator candidate, with the second-highest final accuracy score, close to that of Claude Opus 4.7 without thinking mode. Although caution is needed because of its high excess OK rate, we evaluated the same hypothesis–verification pairs as those evaluated in the main text using GPT-5.5.

The evaluation results obtained using GPT-5.5 are shown in Table 6. Compared with Table 2, the final judgment scores are generally higher, which is consistent with the high excess OK rate. Among the individual evaluation criteria, criterion

(e) showed a large discrepancy from the Claude Opus 4.7 evaluation shown in Table 2.

We note that GPT-5.5 assigned a high score to GPT-5.5 itself, giving it the highest score in its own evaluation. In contrast, the Claude Opus 4.7 evaluation did not assign the highest score to Claude Opus 4.7 itself. However, Claude Opus 4.7 scored itself higher than Gemini 3.1 Pro, whereas GPT-5.5 scored Gemini 3.1 Pro higher than Claude Opus 4.7. For the other four models, excluding Claude Opus 4.7 and GPT-5.5, the ranking remained the same: GPT-5.5 Pro, Gemini 3.1 Pro, DeepSeek-R1, and Qwen3 235B A22B 2507. This difference may reflect differences in the interpretation of the criteria for what constitutes appropriate hypotheses and verification methods, rather than simple model preference.

These results show that, although the evaluation results were consistent when the evaluator models themselves were excluded, evaluator models may exhibit a tendency toward self-favorable bias. Therefore, using an independent model is preferable for objective evaluation.

D Evaluation of Small Models

We performed evaluations of GPT-5.4 nano (OpenAI, 2026a) and Gemma 3 IT 4B (Gemma Team et al., 2025) as demonstrations of how the HYDRA framework can be used to evaluate smaller models. The experimental setup, situation datasets, API-based model access, prompts, and evaluator model, Claude Opus 4.7, were the same as those described in the main text. We note that two hypothesis–verification pairs generated by Gemma 3 IT 4B were refused evaluation by Claude Opus 4.7 and were therefore excluded from the score calculations.

The results are summarized in Table 7. We found that the scores were significantly lower than those of the models evaluated in the main text, especially for Gemma 3 IT 4B. This indicates that the task defined in the HYDRA framework is not trivial and may be useful for evaluating models of various sizes.

Model	(a)	(b)	(c)	(d)	(e)	(f)	Final judge
GPT-5.5	99.6%	**99.6%**	98.1%	**99.6%**	**99.6%**	98.1%	**95.9%**
GPT-5.5 Pro	**100%**	**99.6%**	98.5%	**99.6%**	97.8%	**98.5%**	95.5%
Gemini 3.1 Pro	**100%**	99.3%	99.6%	98.5%	97.4%	98.1%	94.1%
Claude Opus 4.7	99.3%	98.5%	98.9%	98.9%	94.4%	97.8%	91.1%
Qwen3 235B A22B 2507	97.4%	96.7%	98.9%	98.1%	96.3%	96.7%	89.2%
DeepSeek-R1	97.0%	98.1%	**100%**	98.1%	96.3%	98.1%	90.3%
Average	98.9%	98.6%	99.0%	98.8%	97.0%	97.9%	92.7%

Table 6: Success rates for hypothesis and verification method formulation across the six tested LLMs, evaluated by GPT-5.5, in the same format as Table 2. Bold numbers indicate the best score for each criterion and the final result.

Model	(a)	(b)	(c)	(d)	(e)	(f)	Final judge
GPT-5.4 nano	95.9%	91.8%	100%	97.4%	84.4%	100%	73.6%
Gemma 3 IT 4B	80.5%	69.3%	99.6%	70.4%	74.5%	99.3%	36.0%

Table 7: Success rates for hypothesis and verification method formulation for the two small models, presented in the same format as Table 2.