RuleWeaver: Benchmarking Rule-Centered Scenario Reasoning for Large Language Models

Bohan Yu[1,2*], **Shi-Yang Li**[1,2*], **Pengfei Cao**[2,3†], **Jun Zhao**[2,3], **Kang Liu**[2,3†]
[1]School of Advanced Interdisciplinary Sciences, University of Chinese Academy of Sciences
[2]The Key Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences
[3]School of Artificial Intelligence, University of Chinese Academy of Sciences
{yubohan2025,lishiyang2026}@ia.ac.cn {pengfei.cao,jzhao,kliu}@nlpr.ia.ac.cn

Abstract

Large language models (LLMs) are increasingly applied to specialized domains, where effective use of domain expertise often requires reasoning over complex rules in concrete scenarios. However, existing benchmarks only partially evaluate this capability, as they either focus on output-level instruction constraints or overlook the distinct roles that rules play in scenario reasoning. To address these gaps, this paper introduces **RuleWeaver**, a benchmark construction framework for evaluating rule-centered scenario reasoning. RuleWeaver starts from corpus-derived IF-THEN *Meta Rules*, progressively augments them into complex rules, and composes these rules into rule-centered scenario QA instances. Beyond final-answer correctness, RuleWeaver further supports process-level evaluation through rubric-based answer quality, rule recall, and rule precision. Experiments on 11 representative LLMs show that current models still struggle with complex rule-centered scenario reasoning, with even the best-performing model achieving only around 50% of the maximum rubric score. We make our code and dataset available here: RuleWeaver.

1 Introduction

As large language models (LLMs) demonstrate strong capabilities in text understanding, reasoning, and generation (OpenAI et al., 2024; Zhao et al., 2026), applying them in specialized domains has become increasingly important and urgent. In such domains, it is essential to effectively leverage domain expertise, especially when this expertise is represented as explicit rules (Guha et al., 2023; Shen et al., 2024). For example, under a refund policy, a model may need to decide whether a customer can receive a refund by checking rules such as "*items can be refunded within 7 days*", "*opened items cannot be refunded*", and "*defective items can still be refunded*". This example illustrates that rules can play different roles in concrete scenarios, including setting conditions, imposing prohibitions, and specifying exceptions. In such contexts, effective model behavior requires more than producing plausible responses: models must identify rules relevant to a concrete scenario, apply them under stated constraints, and justify decisions through explicit rule-based reasoning. We refer to this capability as rule-centered scenario reasoning.

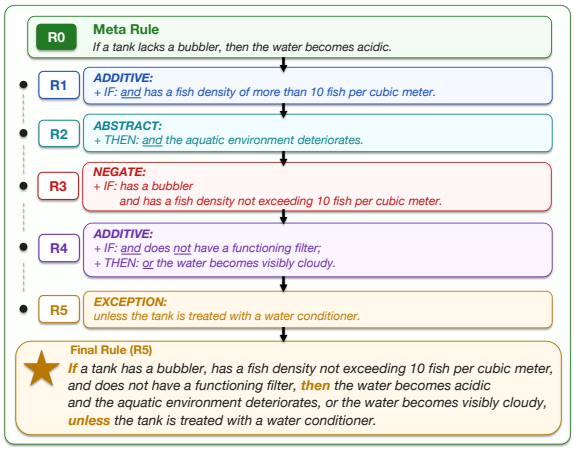


Figure 1: Progressive rule construction: an example of transforming a WikiHow (Koupaee and Wang, 2018) Meta Rule into a complex rule through five-round semantic enhancement.

However, existing benchmarks, such as those for instruction following and logical reasoning, capture only part of this capability. In specific, instruction-following datasets (Jiang et al., 2024; Xu et al., 2024; Qin et al., 2024; Sun et al., 2025) evaluate whether models can understand constraints embedded in natural-language instructions, such as output format and content style. Although these constraints may be expressed in different ways, they primarily prescribe what outputs should be produced or presented rather than guide reasoning or decision-making processes. In contrast, logic-

* Equal contribution.
† Corresponding authors.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 10854–10878
October 24-29, 2026 ©2026 Association for Computational Linguistics

reasoning datasets (Saparov and He, 2023; Han et al., 2024; Parmar et al., 2024; Patel et al., 2024; Zhou et al., 2025; Qi et al., 2025) evaluate the capability required by rule-centered scenario reasoning under the condition that the relevant rules are provided in advance. However, they mainly evaluate whether models can derive conclusions from the given rules, without sufficiently distinguishing the different roles that rules may play in realistic scenarios, such as exceptions, conflicts, or priorities. Consequently, current benchmarks provide limited support for evaluating whether LLMs can identify relevant rules, understand their roles and dependencies, and apply them to answer questions grounded in concrete scenarios.

To address these gaps, this paper introduces **RuleWeaver**, a benchmark construction framework for evaluating LLMs on complex rule-centered scenario reasoning. The basic idea is to address two challenges: (1) building fine-grained rules with different roles, and (2) generating concrete scenarios that require those rules to be identified and applied. Specifically, for the first challenge, RuleWeaver starts from *Meta Rules*, defined as atomic IF-THEN rules with a single condition and a single outcome, extracted from diverse real-world corpora spanning policy reports, procedural guides, contractual clauses, and narrative contexts. This design ensures that the Meta Rules are derived from naturally occurring rule expressions across different domains. RuleWeaver then progressively augments each Meta Rule into a *rule group* of complex variants through six semantic types: **ABSTRACT**, **ADDITIVE**, and **NEGATE** capture underspecification, supplementary constraints, and semantic reversal, while **EXCEPTION**, **CONFLICT**, and **IRONCLAD** introduce applicability changes, incompatible implications, and priority constraints. It enables fine-grained evaluation of whether models can understand and distinguish different rules. Figure 1 illustrates one such augmentation path: starting from a Meta Rule (R0), RuleWeaver applies sequential semantic enhancements to produce a complex rule (R5), with each step modifying the condition side, the outcome side, or both.

To address the second challenge, RuleWeaver converts the aforementioned rule groups into rule-centered scenario QA instances spanning seven predefined question types. Each instance is built around five relevant complex rules sampled from distinct rule groups under either same-source or cross-source settings. The construction mainly includes four stages: (1) *dependency planning* selects the target question type and builds a rule dependency graph that specifies the rule applied at each reasoning step; (2) *sub-scenario generation* instantiates each planned step as a local scenario fragment with the facts needed to trigger the corresponding rule; (3) *final synthesis* merges sub-scenarios into a coherent scenario with a final question and reference outputs, including rule application annotations, a rule logic chain, and an instance-specific rubric; and (4) *iterative quality review* checks structural validity, rule coverage, information leakage, and dependency consistency. This process ensures that the QA instances are rule-centered, with each scenario grounded in rule applications and rule-dependency chains. Overall, RuleWeaver constructs 200 rule groups covering six semantic types and 96 scenario QA instances spanning seven question types.

For evaluation, RuleWeaver goes beyond final-answer correctness by using process annotations and instance-specific rubrics. Rule recall and rule precision measure whether models can identify relevant rules and apply them correctly, while rubrics assess answer quality across multiple dimensions of rule-centered scenario reasoning. Under this protocol, we evaluate 11 representative LLMs. The best rubric scores reach only 53.83 in the same-source setting and 50.27 in the cross-source setting, while the weakest rubric dimension achieves only a 14.2% normalized score, revealing a clear gap between current LLM capabilities and reliable rule-centered scenario reasoning. Further analyses show that models often fail not only to identify all relevant rules, but also to apply them correctly and track dependencies among rule applications.

Our contributions are twofold:

- This paper introduces RuleWeaver, a benchmark construction framework for evaluating rule-centered scenario reasoning. It builds fine-grained complex rules from corpus-derived Meta Rules and composes them into rule-centered scenario QA instances.

- RuleWeaver supports process-level evaluation through rubric-based answer quality, rule recall, and rule precision, revealing key weaknesses in LLMs' rule-centered scenario reasoning capabilities.

Dataset	Corpus-Derived Rules	Semantic Rule Typing	Rule-Centered Scenario QA with Process-Level Scoring	Multi-Source / Cross-Domain Composition
Instruction-Following Benchmarks				
IFEval (Zhou et al., 2023)	✗	✗	✗	✗
WizardLM (Xu et al., 2024)	✗	✗	✗	✗
InfoBench (Qin et al., 2024)	✗	✗	✗	✗
FollowBench (Jiang et al., 2024)	✗	✗	✗	✗
ComplexBench (Wen et al., 2024)	✗	✗	✗	✗
Logic-Reasoning Benchmarks				
RuleTaker (Clark et al., 2020)	✗	✗	⊖	✗
AR-LSAT (Zhong et al., 2021)	⊖	⊖	✗	✗
ProofWriter (Tafjord et al., 2021)	✗	⊖	⊖	✗
PrOntoQA (Saparov and He, 2023)	✗	⊖	⊖	✗
FOLIO (Han et al., 2024)	✗	⊖	✗	✗
LogicBench (Parmar et al., 2024)	✗	⊖	⊖	✗
Multi-LogiEval (Patel et al., 2024)	✗	⊖	⊖	✗
RuleArena (Zhou et al., 2025)	✓	✗	✓	⊖
ProverQA (Qi et al., 2025)	✗	⊖	⊖	✗
RuleWeaver (Ours)	✓	✓	✓	✓

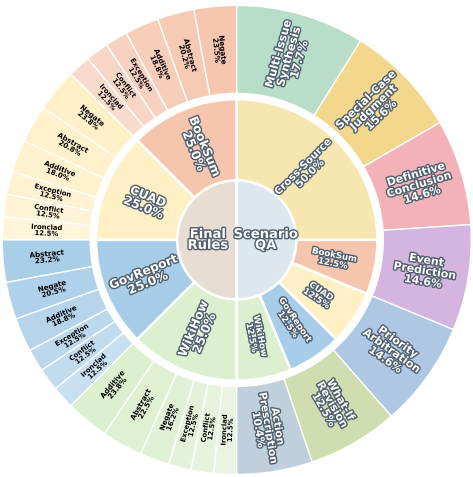


Table 1: **Left:** comparison of RuleWeaver with instruction-following and logical reasoning benchmarks across core evaluation dimensions. ✓: support; ⊖: partial support; ✗: unsupported. **Right:** distribution of the 200 complex rule groups and 96 scenario QA instances. Detailed rule-source and QA-type distributions are provided in Appendices A.1 and A.2, respectively, with question-type descriptions in Appendix B.4.

2 Related Work

Rule-centered scenario reasoning is closely related to two lines of LLM evaluation: instruction-following benchmarks and logical reasoning benchmarks. Table 1 (left) summarizes representative datasets along these dimensions.

Instruction-Following Benchmarks Instruction-following benchmarks mainly evaluate whether models can satisfy constraints embedded in natural language instructions, such as format and style requirements (Jiang et al., 2024; Xu et al., 2024; Qin et al., 2024; Zhou et al., 2023; Wen et al., 2024). These constraints can be viewed as weak rule-like requirements, and some benchmarks further decompose or compose them to evaluate multi-constraint compliance (Jiang et al., 2024; Wen et al., 2024). However, such benchmarks usually treat rules as output constraints for compliance checking, rather than as reasoning units with different roles and condition-triggered consequences. As a result, they provide limited support for analyzing rule roles, such as abstraction, negation, exception, conflict, and non-overridable constraints, or for evaluating how different types of rules are selected and applied in concrete scenarios.

Logical Reasoning Benchmarks Logical reasoning benchmarks (Clark et al., 2020; Tafjord et al., 2021; Saparov and He, 2023; Han et al., 2024; Parmar et al., 2024; Yu et al., 2026) explicitly use rules as premises for inference. Some further introduce formal logic annotations, controllable reasoning depth, or real-world rules (Patel et al., 2024; Zhou et al., 2025; Qi et al., 2025). However, as shown in Table 1 (left), process-level scoring and multi-source or cross-domain rule composition remain underexplored in most existing benchmarks. More importantly, these benchmarks primarily assess whether models can apply provided rules to derive conclusions, often treating rules as homogeneous inference premises. This makes it difficult to evaluate whether models can distinguish different rule roles, identify rules relevant to a scenario, and combine multiple rules through scenario-specific dependencies. As a result, existing benchmarks do not fully capture the rule selection, role understanding, and dependency-aware application required by rule-centered scenario reasoning.

RuleWeaver bridges these lines by constructing complex rules with different roles from corpus-derived Meta Rules and composing them into concrete QA scenarios. This enables fine-grained evaluation of rule selection, role understanding, dependency-aware application, and process-level answer quality.

3 RuleWeaver

RuleWeaver is a traceable benchmark construction framework for rule-centered scenario reasoning, built from corpus-derived *Meta Rules*. A Meta Rule is an atomic IF-THEN rule with one triggering condition and one outcome, serving as the minimal unit for progressive rule augmentation. As shown in Figure 2, RuleWeaver transforms these units into complex rule groups and composes them into rule-centered scenario QA instances.

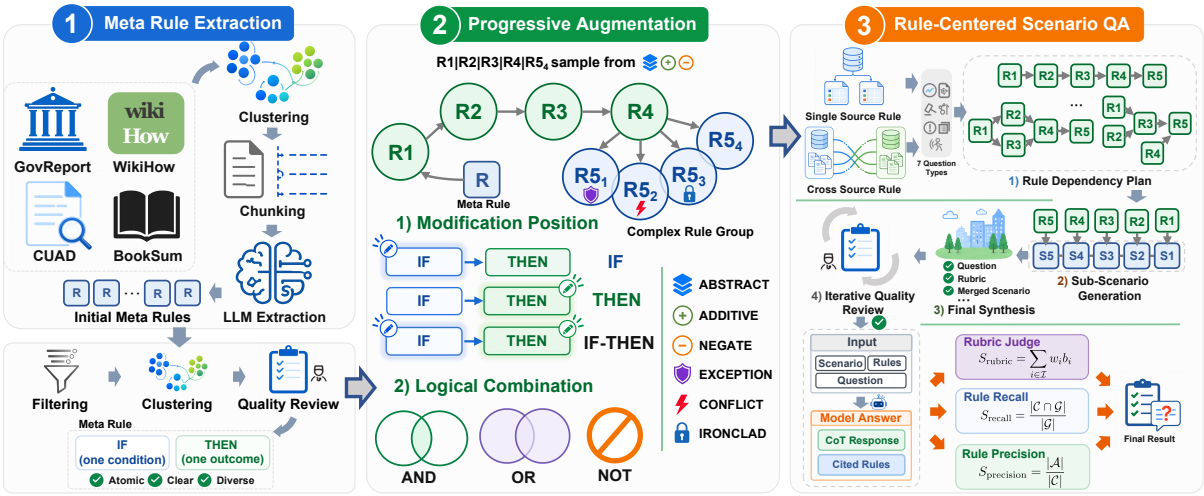


Figure 2: Overview of RuleWeaver: from real-world corpora to progressively augmented complex rules, rule-centered scenario QA, and process-level scoring.

3.1 Rule Augmentation

Rule augmentation defines the semantic design space for transforming a Meta Rule into a more complex variant. We represent each augmentation with three attributes: semantic enhancement type, modification position, and logical combination method. The semantic enhancement type captures the intended rule change and includes six categories: **ABSTRACT**, **ADDITIVE**, **NEGATE**, **EXCEPTION**, **CONFLICT**, and **IRONCLAD**. The first three introduce moderate changes to rule granularity, scope, or polarity, while the latter three model higher-impact rule interactions involving applicability, compatibility, or priority. The modification position records whether the change applies to the IF side, the THEN side, or both. The logical combination method records how the added information is composed with the original rule through AND, OR, or NOT. Together, these attributes make each augmentation explicit and traceable, enabling controlled variation in rule complexity and semantic type. Detailed definitions and representative examples are provided in Appendix B.1.

3.2 Complex Rule Construction

We construct complex rule groups in three stages: Meta Rule extraction, quality filtering, and progressive augmentation. Prompts and implementation details for these stages, including generation and filtering models, are provided in Appendix B.2.

Initial Meta Rule Construction We extract Meta Rules from real corpus documents so that the basic rule units reflect naturally occurring constraint expressions. We use four sources: GovReport (Huang et al., 2021), WikiHow (Koupaee and Wang, 2018), CUAD (Hendrycks et al., 2021), and BookSum (Kryscinski et al., 2022), covering policy reports, procedural guides, contractual clauses, and narrative contexts; examples are shown in Appendix B.3. To improve source coverage, we cluster documents within each corpus and sample representative documents for rule extraction, yielding 11,145 initial Meta Rules.

Meta Rule Quality Filtering We filter the extracted rules for format validity, atomicity, semantic clarity, and independence from external background knowledge. To improve diversity and reduce redundancy, we further perform representative clustering and manual inspection, resulting in 200 high-quality Meta Rules. These Meta Rules cover diverse sources and expression styles while preserving clear atomic IF-THEN structure.

Progressive Complex Rule Construction Starting from the 200 filtered Meta Rules, we construct complex rules through a five-round progressive augmentation process. For each Meta Rule, the first four rounds form a shared augmentation trajectory, where one moderate enhancement type is sampled from ABSTRACT, ADDITIVE, and NEGATE and applied to the IF side, THEN side, or both. Because EXCEPTION, CONFLICT, and IRONCLAD can substantially affect rule applicability, priority, or rule interactions, they are reserved for the fifth round, where each trajectory branches into four final variants: one sampled from ABSTRACT, ADDITIVE, and NEGATE, and one for each high-impact type. This yields 200 complex rule groups

whose variants collectively cover all six enhancement types, while preserving traceability to the initial Meta Rule, augmentation rounds, modification positions, and logic combinations.

3.3 Rule-Centered Scenario QA Construction

After constructing grouped complex rules, we generate rule-centered scenario QA instances through four stages: dependency planning, sub-scenario generation, final synthesis, and quality review. We use DeepSeek-V4-Pro (DeepSeek-AI et al., 2026) throughout all stages and consider two QA settings: *same-source* QA, where five relevant rules are sampled from a single source dataset, and *cross-source* QA, where the rules are sampled across four datasets. For each instance, the rule set $\mathcal{R} = \{r_1, \ldots, r_5\}$ is sampled from distinct rule groups to avoid including sibling variants of the same Meta Rule, and each rule is assigned a unique identifier for dependency tracking. Prompts covering all stages are shown in the Figures 22– 27.

Dependency Planning Given $\mathcal{R}$, the generator first samples a target question type from seven predefined categories, covering multi-issue synthesis, definitive conclusion derivation, event prediction, priority arbitration, special-case judgment, what-if revision, and action prescription (see Appendix B.4 for detailed definitions). It then constructs a *rule dependency plan*, a directed structure over rule-application steps that specifies which rule is applied at each step and how intermediate rule-derived conclusions support later steps. This plan controls the reasoning topology of each QA instance, including independent root rules, sequential dependency transfer, fan-out branching, convergence over multiple rule conclusions, and multi-level aggregation. By explicitly encoding these dependencies, RuleWeaver can generate scenarios that require models not only to apply individual rules, but also to preserve and combine intermediate conclusions across the rule chain.

Sub-Scenario Generation The generator then instantiates each dependency step as a local *sub-scenario*. For each step, it observes the current rule, the already generated predecessor steps, and the next downstream step to which the current step feeds, together with its associated rule. It then writes a scenario fragment together with auxiliary fields used only during construction. These fields include trigger facts, notes on facts that trigger exception or conflict rules, an intermediate conclusion, and the conclusion's downstream contribution. The fragment embeds rule-triggering facts through natural contextual details, while avoiding explicit disclosure of rule text, rule identifiers, hidden sub-questions, or rule conclusions.

Final Synthesis Finally, the enriched steps are merged into a single coherent scenario and one final question. The generator also produces the rule-application annotations for computing rule-application accuracy, a *rule logic chain*, and a QA-specific all-or-nothing *rubric*. The rule logic chain serves as the reference reasoning path for rubric-based scoring by recording, for each reasoning step, the rule identifier, dependencies, trigger facts, intermediate conclusion, and downstream contribution. The rubric defines the scoring criteria for each instance, covering aspects such as logical alignment, dependency transfer, exception/conflict handling, and final-answer correctness.

Quality Review and Refinement The generated QA is further refined through an iterative quality-control loop, typically run for five rounds. After a candidate instance is produced, heuristic checks verify basic validity, such as well-formed JSON outputs and coverage of directly used rules. LLM-based judges further assess the dependency plan, sub-scenarios, and final synthesis for structural validity, rule coverage, information leakage, and dependency transfer. The identified issues are summarized into stage-specific feedback and used to optimize the scenario in subsequent refinement rounds. Instances whose review scores exceed the predefined quality threshold of 80 points are accepted and subsequently subjected to human inspection. This loop enables later iterations to produce scenarios with clearer dependencies, better rule coverage, and less information leakage.

3.4 Scoring Design

During evaluation, each model receives a final scenario, a corresponding question, and a visible rule pool containing both relevant and distractor rules, then returns a final answer together with the rule identifiers it cites or applies. We report three complementary scores: rubric-based answer quality, rule recall, and rule precision. For answer quality, the LLM judge applies the instance-specific all-or-nothing rubric. The rubric has a total value of 100 points, distributed across its dimensions. Let $\mathcal{I}$ denote the rubric dimensions, let w_i be the point value of dimension i with $\sum_{i \in \mathcal{I}} w_i = 100$,

and let $b_i \in \{0, 1\}$ indicate whether the model answer satisfies dimension i. A satisfied dimension receives its full point value; otherwise it receives zero. The rubric score is $S_{\text{rubric}} = \sum_{i \in \mathcal{I}} w_i b_i$. For rule retrieval, let $\mathcal{G}$ be the set of gold relevant rule identifiers and $\mathcal{C}$ be the set of identifiers cited by the model. Rule recall is $S_{\text{recall}} = \frac{|\mathcal{C} \cap \mathcal{G}|}{|\mathcal{G}|}$. For rule application, let $\mathcal{A}$ denote the cited rule applications judged correct with respect to the reference answer. Rule precision is $S_{\text{precision}} = \frac{|\mathcal{A}|}{|\mathcal{C}|}$, with $S_{\text{precision}}$ set to zero when no rule is cited. Reporting these three scores separately distinguishes answer quality, rule retrieval, and rule application.

3.5 Human Quality Assurance

Human quality assurance supplements RuleWeaver's automatic filtering and LLM-based review. Reviewers assess Meta Rules for atomicity, clarity, source faithfulness, and domain relevance; complex rules for traceability, semantic-type correctness, and logical consistency; and scenario QA items for coherent scenarios, questions, rule chains, annotations, and rubrics. Items that are ambiguous, inconsistent, insufficiently rule-grounded, or prone to information leakage are revised or excluded.

4 Evaluation

4.1 Model Selection

We evaluate 11 state-of-the-art models spanning multiple model families on 96 complex scenario QA instances: Claude-Opus-4.6 (Anthropic, 2026a), Claude-Sonnet-4.6 (Anthropic, 2026b), Deepseek-V4-Pro (DeepSeek-AI et al., 2026), Qwen3.5-Plus (Qwen Team, 2026), GPT-5.4 (OpenAI, 2026b), GPT-5.5 (OpenAI, 2026a), Gemini-3.1-Pro-Preview (Google, 2026), Doubao-Seed-2.0-pro (ByteDance Seed, 2026), GLM-5 (GLM-5-Team et al., 2026), Kimi-K2.6 (Moonshot AI, 2026), and MiniMax-M2.7 (MiniMax, 2026).

4.2 Evaluation Settings

For all evaluated models, we disable reasoning modes when configurable and set the decoding temperature to 0 to ensure deterministic generation, except for Kimi-K2.6, whose APIs require the temperature to be set to 0.6 for non-thinking mode. For Gemini-3.1-Pro-Preview, we set the thinking configuration to low. We use DeepSeek-V4-Flash (DeepSeek-AI et al., 2026) as the judge model, with its temperature also set to 0. Each instance contains five gold rules, and our main evaluation is conducted under the settings with 200 complex rules. Rule recall is computed by detecting the rule identifiers explicitly cited in each model response and comparing them with the reference set of required rules. The QA template and judge prompt are provided in Figures 28 and 29, respectively.

4.3 Main Results

Overall Performance Table 2 reports the main 200-rule results. Under same-source composition, GPT-5.5 obtains the highest rubric score (53.83) and rule precision (74.58), while Kimi-K2.6 achieves the highest rule recall (72.92). Under cross-source composition, Claude-Opus-4.6 is strongest in both rubric score (50.27) and rule recall (62.50), whereas GPT-5.4 obtains the highest rule precision (78.64). Averaged over the 11 models, cross-source evaluation reduces recall by 11.93 points and rubric score by 4.05 points, while precision increases slightly by 2.42 points. This indicates that heterogeneous rule composition primarily stresses complete rule retrieval and rule-centered scenario reasoning. Bootstrap analyses further confirm that the reported point estimates are stable under resampling, with detailed results provided in Appendix C.1.

Semantic Capabilities The upper radar in Table 2 analyzes performance by semantic augmentation type, with detailed numerical results provided in Appendix C.2. We score semantics at the enhancement-event level, so a rule may contribute multiple events when its five-round enhancement contains repeated semantic types. For a model m and semantic type s, the semantic score is the fraction of events in $E(m, s)$ whose parent rule is judged as a correct application, i.e., $\sum_{e \in E(m,s)} \mathbf{1}[\text{rule}(e) \in \mathcal{A}(e)]/|E(m, s)|$, where $\mathcal{A}(e)$ is the set of correctly applied rules in the judge output for the answer containing event e. Overall, CONFLICT is the strongest semantic category (42.7%), followed by IRONCLAD (40.6%) and ABSTRACT (39.7%), while EXCEPTION is the weakest (34.6%). At the model level, Claude-Opus-4.6 leads five of the six semantic axes, while Claude-Sonnet-4.6 leads EXCEPTION. The consistently low EXCEPTION score suggests that models remain especially brittle when rule applicability is conditioned on exception structure rather than direct condition matching.

Model	Same-Source			Cross-Source		
	Recall	Precision	Rubric	Recall	Precision	Rubric
Claude-Opus-4.6	70.42	70.41	48.06	**62.50**$_{\downarrow 7.92}$	73.74$_{\uparrow 3.33}$	**50.27**$_{\uparrow 2.21}$
Claude-Sonnet-4.6	71.67	63.15	43.96	59.58$_{\downarrow 12.09}$	54.74$_{\downarrow 8.41}$	36.56$_{\downarrow 7.40}$
GPT-5.4	61.67	71.31	48.21	48.33$_{\downarrow 13.34}$	**78.64**$_{\uparrow 7.33}$	42.58$_{\downarrow 5.63}$
GPT-5.5	69.17	**74.58**	**53.83**	52.08$_{\downarrow 17.09}$	69.48$_{\downarrow 5.10}$	46.40$_{\downarrow 7.43}$
Gemini-3.1-Pro-Preview	60.83	58.44	31.77	45.83$_{\downarrow 15.00}$	72.95$_{\uparrow 14.51}$	27.81$_{\downarrow 3.96}$
Deepseek-V4-Pro	63.75	59.22	34.69	55.00$_{\downarrow 8.75}$	60.10$_{\uparrow 0.88}$	28.38$_{\downarrow 6.31}$
Qwen3.5-Plus	66.67	66.32	37.75	55.83$_{\downarrow 10.84}$	64.27$_{\downarrow 2.05}$	34.15$_{\downarrow 3.60}$
GLM-5	62.50	64.17	37.54	51.67$_{\downarrow 10.83}$	69.21$_{\uparrow 5.04}$	32.35$_{\downarrow 5.19}$
Doubao-Seed-2.0-pro	48.33	61.63	27.67	39.58$_{\downarrow 8.75}$	75.73$_{\uparrow 14.10}$	33.02$_{\uparrow 5.35}$
MiniMax-M2.7	65.00	47.43	29.69	52.08$_{\downarrow 12.92}$	43.06$_{\downarrow 4.37}$	21.15$_{\downarrow 8.54}$
Kimi-K2.6	**72.92**	55.13	32.50	59.17$_{\downarrow 13.75}$	56.53$_{\uparrow 1.40}$	28.48$_{\downarrow 4.02}$

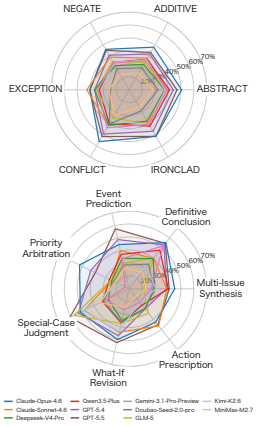


Table 2: **Left:** Main results under the 200 complex-rule setting. Rule recall and precision are percentages, rubric score is on a 0–100 scale, arrows indicate cross-source changes relative to same-source scores, and bold marks the best result in each column. **Right:** Radar summaries of semantic correct-application performance (top) and seven question-type performance (bottom).

Performance Across Question Types The lower radar summarizes performance across the seven scenario question types under the same 200-rule setting, with detailed question-type definitions and numerical results provided in Appendices B.4 and C.3. Across models, average rubric scores are highest for Special-Case Judgment (43.5) and Definitive Conclusion (43.4), whereas Priority Arbitration (28.4) and Action Prescription (31.8) are more difficult. The leading model also varies by question type: GPT-5.5 performs best on Event Prediction, Special-Case Judgment, and What-If Revision; Claude-Opus-4.6 leads Multi-Issue Synthesis, Definitive Conclusion, and Priority Arbitration; and Claude-Sonnet-4.6 leads Action Prescription. These differences suggest that model capability is not monolithic, but instead depends on the kind of reasoning a scenario demands.

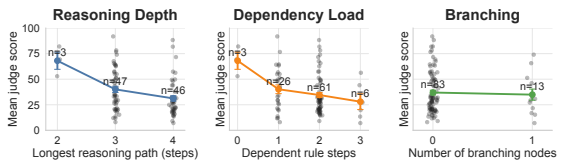
Rubric-Dimension Performance To analyze rubric-level performance loss, we average normalized dimension scores over all model–QA pairs; dimension definitions are provided in Appendix B.5. Table 3 shows that models perform well on surface-level constraints, including No External Facts (91.2%) and Citation Format Compliance (82.8%). By contrast, the weakest dimensions are Dependency Chain Alignment (14.2%), Intermediate Conclusion Quality (21.0%), Issue Decomposition (28.4%), and Exception/Conflict Handling (30.4%). This pattern suggests that the central bottleneck is maintaining rule-grounded intermediate reasoning across dependent steps, rather than response formatting or unsupported factual additions. A model-level heatmap is provided in Appendix C.4. Additional QA-level correlation analyses and qualitative case studies are provided in Appendices C.5 and C.6.

Rubric Dimension	Max	Mean	Norm.
Question Understanding	10	5.28	52.8%
Issue Decomposition	10	2.84	28.4%
Citation Format Compliance	8	6.62	82.8%
Rule-Grounded Reasoning	17	6.44	37.9%
Dependency Chain Alignment	18	2.55	14.2%
Exception/Conflict Handling	12	3.65	30.4%
Intermediate Conclusion Quality	10	2.10	21.0%
Final Answer Consistency	12	4.36	36.3%
No External Facts	3	2.74	91.3%

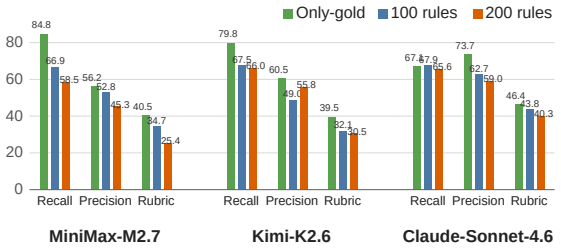
Table 3: Average rubric-dimension scores under the 200-rule setting. "Norm." denotes the mean score normalized by each dimension's maximum points.

4.4 Analysis

Scenario Complexity Analysis We analyze scenario complexity from each instance's *rule logic chain*, treating reasoning steps as nodes and dependency annotations as directed edges. Let $\mathcal{V}$ be the set of all reasoning nodes, $\mathcal{V}_{\text{rule}} \subseteq \mathcal{V}$ the rule-application nodes, $\mathcal{P}$ the set of directed paths, and $\deg^-(v)$ and $\deg^+(v)$ the in- and out-degree of node v. We use three graph metrics: reasoning depth $D = \max_{p \in \mathcal{P}} |p|$, where $|p|$ counts the number of reasoning steps on path p; dependency load $L = \sum_{v \in \mathcal{V}_{\text{rule}}} \mathbf{1}[\deg^-(v) > 0]$, the number of rule steps that depend on earlier conclusions; and structural branching $B = \sum_{v \in \mathcal{V}} \mathbf{1}[\deg^+(v) > 1]$, the number of nodes feeding multiple downstream steps. Figure 3a shows that reasoning depth is the strongest complexity signal. Mean score decreases



(a) QA-level performance by scenario complexity.



(b) Overall performance under different rule-pool settings.

Figure 3: **Top**: gray points show individual QA instances, and colored markers show mean rubric scores with standard-error bars across scenario complexity levels. **Bottom**: performance under only-gold, 100-rule, and 200-rule settings for three selected models.

from 68.1 at depth 2 to 40.0 at depth 3 and 31.2 at depth 4, indicating that longer rule chains are substantially harder for current models. Dependency load shows a similar trend: scores decline from 68.1 with no dependent rule step to 28.0 with three dependent rule steps. In contrast, structural branching has little visible association with performance, with branch-free and one-branch scenarios showing similar mean scores. Overall, sequential depth and the need to reuse intermediate conclusions emerge as the main structural difficulty factors in the current benchmark.

Rule-Pool Size Sensitivity To assess sensitivity to rule-pool size, we evaluate three settings: only-gold rules, 100 rules, and 200 rules. Figure 3b reports recall, precision, and rubric score for MiniMax-M2.7, Kimi-K2.6, and Claude-Sonnet-4.6. Performance generally declines as the rule pool expands. MiniMax-M2.7 shows the clearest degradation, with recall dropping from 84.8 to 58.5, precision from 56.2 to 45.3, and rubric score from 40.5 to 25.4. Kimi-K2.6 also drops substantially in recall and rubric score, although its precision partially recovers from the 100-rule setting while remaining below only-gold. Overall, larger rule pools mainly increase the distractor burden: models must identify the relevant rules among more candidates, and final score further depends on whether the selected rules are correctly applied and integrated into the reasoning chain.

Error Type	All		Full Recall	
	Count	Rate	Count	Rate
Rule selection	762	72.2%	–	–
Cited-rule application	580	54.9%	191	65.6%
Rule interaction	734	69.5%	166	57.0%
Multi-step integration	916	86.7%	218	74.9%
Final synthesis	678	64.2%	193	66.3%

Table 4: Error-type prevalence over all model–QA answers and the subset with full rule recall.

Error Type Analysis To complement rubric-dimension scores, we group failures into five rule-centered error types using existing evaluation annotations. Rule selection errors are identified from missed gold rules in rule recall; cited-rule application errors from rule precision annotations; rule interaction errors from exception/conflict handling annotations; multi-step integration errors from issue decomposition, intermediate-conclusion, and dependency-chain annotations; and final synthesis errors from final-answer consistency annotations. Because an answer can fail in multiple ways, the analysis is multi-label. As shown in Table 4, the most frequent error is multi-step integration (86.7%), followed by rule selection (72.2%) and rule interaction (69.5%). In the full-recall subset, multi-step integration errors (74.9%) and cited-rule application errors (65.6%) remain frequent, suggesting that models often fail not only by missing relevant rules, but also by misusing retrieved rules or failing to connect intermediate conclusions.

Human-Judge Agreement To assess the reliability of rubric-based automatic evaluation, we compare LLM-judge total scores with human judgments on score-stratified samples. Detailed sampling and annotation procedures are provided in Appendix B.6. Table 5 reports agreement on total rubric scores. The GPT-5.5 audit yields Spearman's $\rho = 0.808$ (Spearman, 1904) and Pearson's $r = 0.795$ (Pearson, 1895) between human and LLM-judge scores. Across the four additional audits, judge–human Spearman correlations range from 0.755 to 0.842 and Pearson correlations from 0.635 to 0.826; human–human Spearman correlations range from 0.935 to 0.977 and Pearson correlations from 0.901 to 0.969. These results indicate strong agreement between human and automatic scoring, supporting the reliability of rubric-based evaluation.

GPT-5.5 Audit	Additional Cross-Model Audits				
	Audit outputs	**Judge–human ρ**	**Judge–human r**	**Human–human ρ**	**Human–human r**
Judge–human agreement	Claude-Opus-4.6	0.755	0.733	0.977	0.969
Spearman's ρ: 0.808	Claude-Sonnet-4.6	0.842	0.826	0.945	0.952
Pearson's r: 0.795	DeepSeek-V4-Pro	0.795	0.635	0.947	0.901
	GPT-5.4	0.834	0.762	0.935	0.902

Table 5: Agreement on total rubric scores. Each audit contains 48 answers. ρ: Spearman; r: Pearson.

Judge	Setting	Spearman ρ	Pearson r
Grok-4.3	Same-source	0.882	0.878
	Cross-source	0.918	0.860
	Overall	0.918	0.893
Gemini-3-Flash	Same-source	0.845	0.818
	Cross-source	0.873	0.793
	Overall	0.882	0.846

Table 6: Correlation of alternative-judge and DeepSeek-V4-Flash model-level mean rubric scores.

Evaluator Robustness Analysis To assess possible model-family bias in automatic scoring, we rescore the same outputs with Grok-4.3 (xAI, 2026) and Gemini-3-Flash (Google, 2025) and compare their model-level mean rubric scores with the DeepSeek-V4-Flash scores. Both alternative judges use temperature 0; Grok-4.3 uses `reasoning_effort=none`, and Gemini-3-Flash uses low reasoning effort. In Table 6, Overall denotes correlations computed from each model's mean rubric score across both same-source and cross-source instances. The high correlations across both source settings and Overall indicate that the main leaderboard patterns are consistent under judges from different model families.

5 Conclusion

This paper presents **RuleWeaver**, a benchmark construction framework that transforms corpus-derived Meta Rules into complex rule-centered scenario QA instances. Beyond final-answer correctness, RuleWeaver evaluates models through rubric-based answer quality, rule recall, and rule precision. Experiments on 11 LLMs show that current models remain limited in complex rule-centered scenario reasoning, especially under cross-source composition, exception and priority reasoning, longer reasoning chains, and larger rule pools. These findings highlight rule-centered scenario reasoning as a key challenge for current LLMs.

Limitations

RuleWeaver remains a controlled benchmark. Its 96 scenario QA instances are built from four English corpora, leaving broader multilingual, domain-specific, and institutional rule settings for future expansion. The current evaluation uses an explicit rule pool and static single-turn QA format, whereas real applications may require retrieval from long documents, interaction with users, or updates to changing rule sets. Future work should extend RuleWeaver to larger rule collections, unstructured-document retrieval, interactive settings, multilingual scenarios, and alternative judging protocols. These extensions could further examine structured and transferable memory for long-horizon interactive reasoning (Liang et al., 2026), cross-lingual consistency of rule representations (Cao et al., 2024), representation-level adaptation for multistep reasoning (Liang et al., 2025), and reasoning-consistent updates to changing, interdependent rule sets (Zhang et al., 2026).

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.62406321) and CIPS-SMP-Zhipu Large Model Fund.

References

Anthropic. 2026a. Claude opus 4.6 system card. https://www-cdn.anthropic.com/14e4fb01875d2a69f646fa5e574dea2b1c0ff7b5.pdf.

Anthropic. 2026b. Introducing Claude Sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6.

ByteDance Seed. 2026. Seed2.0. https://seed.bytedance.com/en/seed2.

Pengfei Cao, Yuheng Chen, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. 2024. One mind, many tongues: A deep dive into language-agnostic knowledge neurons in large language models. *Preprint*, arXiv:2411.17401.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. *Preprint*, arXiv:2002.05867.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, and 300 others. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence. *Preprint*, arXiv:2606.19348.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, and 168 others. 2026. Glm-5: from vibe coding to agentic engineering. *Preprint*, arXiv:2602.15763.

Google. 2025. Gemini 3 flash preview. https://ai.google.dev/gemini-api/docs/models/gemini-3-flash-preview.

Google. 2026. Gemini 3.1 Pro Preview. https://deepmind.google/models/gemini/pro/.

Google AI Blog. 2026. Gemini 3.1 flash-lite: Built for intelligence at scale. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-flash-lite.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Preprint*, arXiv:2308.11462.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alex Wardle-Solano, Hannah Szabo, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. Folio: Natural language reasoning with first-order logic. *Preprint*, arXiv:2209.00840.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *Preprint*, arXiv:2103.06268.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. FollowBench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4667–4688, Bangkok, Thailand. Association for Computational Linguistics.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *Preprint*, arXiv:1810.09305.

Wojciech Kryscinski, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. BOOKSUM: A collection of datasets for long-form narrative summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6536–6558, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sirui Liang, Pengfei Cao, Jian Zhao, Cong Huang, Jun Zhao, and Kang Liu. 2025. Bias-restrained prefix representation finetuning for mathematical reasoning. *Preprint*, arXiv:2511.10707.

Sirui Liang, Pengfei Cao, Jian Zhao, Wenhao Teng, Xiangwen Liao, Jun Zhao, and Kang Liu. 2026. Learning how to remember: A meta-cognitive management method for structured and transferable agent memory. *Preprint*, arXiv:2601.07470.

J. MacQueen. 1967. Some methods for classification and analysis of multivariate observations.

MiniMax. 2026. MiniMax M2.7: Early Echoes of Self-Evolution. https://www.minimax.io/news/minimax-m27-en.

Moonshot AI. 2026. Kimi K2.6: Advancing Open-Source Coding. https://www.kimi.com/blog/kimi-k2-6.

OpenAI. 2025. Gpt-4.1 — mini model documentation. https://developers.openai.com/api/docs/models/gpt-4.1-mini.

OpenAI. 2026a. GPT-5.5. https://openai.com/zh-Hans-CN/index/introducing-gpt-5-5/.

OpenAI. 2026b. Introducing GPT-5.4. https://openai.com/index/introducing-gpt-5-4/.

OpenAI. 2026c. tiktoken: A fast BPE tokenizer for use with OpenAI's models. https://github.com/openai/tiktoken.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. Logicbench: Towards systematic evaluation of logical reasoning ability of large language models. *Preprint*, arXiv:2404.15522.

Nisarg Patel, Mohith Kulkarni, Mihir Parmar, Aashna Budhiraja, Mutsumi Nakamura, Neeraj Varshney, and Chitta Baral. 2024. Multi-logieval: Towards evaluating multi-step logical reasoning ability of large language models. *Preprint*, arXiv:2406.17169.

Karl Pearson. 1895. Vii. note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58(347-352):240–242.

Chengwen Qi, Ren Ma, Bowen Li, He Du, Binyuan Hui, Jinwang Wu, Yuanjun Laili, and Conghui He. 2025. Large language models meet symbolic provers for logical reasoning evaluation. *Preprint*, arXiv:2502.06563.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. InFoBench: Evaluating instruction following ability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13025–13048, Bangkok, Thailand. Association for Computational Linguistics.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *Preprint*, arXiv:2210.01240.

Yongliang Shen, Kaitao Song, Xu Tan, Wenqi Zhang, Kan Ren, Siyu Yuan, Weiming Lu, Dongsheng Li, and Yueting Zhuang. 2024. Taskbench: Benchmarking large language models for task automation. *Preprint*, arXiv:2311.18760.

C. Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

Wangtao Sun, ChenxiangZhang ChenxiangZhang, XueYou Zhang, Xuanqing Yu, Ziyang Huang, Haotian Xu, Shizhu He, Jun Zhao, and Kang Liu. 2025. Beyond instruction following: Evaluating inferential rule following of large language models. In *Proceedings of the 24th China National Conference on Computational Linguistics (CCL 2025)*, pages 1043–1066, Jinan, China. Chinese Information Processing Society of China.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2021. Proofwriter: Generating implications, proofs, and abductive statements over natural language. *Preprint*, arXiv:2012.13048.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. 2024. Benchmarking complex instruction-following with multiple constraints composition. In *Advances in Neural Information Processing Systems*, volume 37, pages 137610–137645. Curran Associates, Inc.

xAI. 2025. Grok 3 beta—the age of reasoning agents. https://x.ai/news/grok-3.

xAI. 2026. Grok 4.3. https://docs.x.ai/developers/models/grok-4.3.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *International Conference on Learning Representations*, volume 2024, pages 30745–30766.

Bohan Yu, Pengfei Cao, Chen Han, Chenxi Zhou, Zhiheng Zhang, Zhiyang Xie, Wenhao Teng, Xiangwen Liao, Jun Zhao, and Kang Liu. 2026. Beyond factual knowledge: Benchmarking and learning step-level procedural rule reasoning in large language models. *Preprint*, arXiv:2608.22753.

Ningyu Zhang, Yunzhi Yao, Jiaxin Qin, Haoming Xu, Yuqi Zhu, Zeping Yu, Mengru Wang, Yuqi Tang, Jia-Chen Gu, Shumin Deng, and Huajun Chen. 2026. Towards principled knowledge editing methods for large language model reasoning. *Nature Machine Intelligence*, 8(8):1189–1200.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2026. A survey of large language models. *Preprint*, arXiv:2303.18223.

Wanjun Zhong, Siyuan Wang, Duyu Tang, Zenan Xu, Daya Guo, Jiahai Wang, Jian Yin, Ming Zhou, and Nan Duan. 2021. Ar-lsat: Investigating analytical reasoning of text. *Preprint*, arXiv:2104.06598.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

Ruiwen Zhou, Wenyue Hua, Liangming Pan, Sitao Cheng, Xiaobao Wu, En Yu, and William Yang Wang. 2025. Rulearena: A benchmark for rule-guided reasoning with llms in real-world scenarios. *Preprint*, arXiv:2412.08972.

Source Corpus	Rule Domain	Groups	Variants	ABSTRACT	ADDITIVE	NEGATE	EXCEPTION	CONFLICT	IRONCLAD
GovReport	Policy reports	50	200	19	16	15	50	50	50
WikiHow	Procedural guides	50	200	21	16	13	50	50	50
CUAD	Contractual clauses	50	200	15	16	19	50	50	50
BookSum	Narrative contexts	50	200	23	15	12	50	50	50
Total	–	**200**	**800**	**78**	**63**	**59**	**200**	**200**	**200**

Table 7: Statistics of the complex final rule set.

(a) QA source, rule-reference, and scenario-length statistics

QA Source	Mode	QAs	Rule Mentions	Avg. Rules	Avg. Words	Median	Word Range
Cross-Source	Cross-source	48	240	5.0	705.1	622.5	241–1,425
BookSum	Same-source	12	60	5.0	796.8	669.5	349–1,499
CUAD	Same-source	12	60	5.0	598.8	584.5	154–1,391
GovReport	Same-source	12	60	5.0	912.1	994.0	289–1,262
WikiHow	Same-source	12	60	5.0	896.0	1,022.0	307–1,476
Total	–	**96**	**480**	**5.0**	**753.0**	**656.5**	**154–1,499**

(b) Question type distribution

Question Type	QAs	Percent
Multi-Issue Synthesis	17	17.71%
Special-Case Judgment	15	15.62%
Definitive Conclusion	14	14.58%
Event Prediction	14	14.58%
Priority Arbitration	14	14.58%
What-If Revision	12	12.50%
Action Prescription	10	10.42%
Total	**96**	**100.00%**

Table 8: Statistics of the scenario QA set, including source balance, rule references, scenario length, and question types. Each scenario QA instance is grounded in five complex rules.

A Data Statistics

A.1 Final Rule Set Statistics

Table 7 summarizes the organized final rule set used by RuleWeaver. The set contains 200 complex rule groups, corresponding to the 200 filtered Meta Rules selected after corpus-specific clustering and manual inspection. The groups are balanced across four source corpora, with 50 groups each from GovReport, WikiHow, CUAD, and BookSum, covering policy reports, procedural guides, contractual clauses, and narrative contexts. Following the progressive construction procedure, each group contains four final variants: one moderate continuation sampled from **ABSTRACT**, **ADDITIVE**, and **NEGATE**, and one variant for each high-impact type, **EXCEPTION**, **CONFLICT**, and **IRONCLAD**. This yields 800 final complex rules in total. The high-impact types are exactly balanced by construction, while the moderate continuation branch contains 78 **ABSTRACT**, 63 **ADDITIVE**, and 59 **NEGATE** variants. The root Meta Rules contain 18.3 words on average, whereas the final complex rule variants contain 71.4 words on average, reflecting the intended accumulation of conditions, outcomes, and semantic interactions.

A.2 Scenario QA Set Statistics

Table 8 summarizes the scenario QA instances generated from the final rule set. The current scenario QA set contains 96 instances, each grounded in exactly five complex rules, yielding 480 rule mentions in total. The data are balanced between 48 cross-source instances and 48 same-source instances. For same-source QA, the four source corpora each contribute 12 instances; for cross-source QA, each instance samples rules across the four corpora, producing a near-balanced rule-source mixture with 59 BookSum, 64 CUAD, 55 GovReport, and 62 WikiHow rule mentions. The final scenarios contain 72,291 words in total, with an average length of 753.0 words per instance (median 656.5; range 154–1,499). Average scenario length ranges from 598.8 words for CUAD-sourced QA to 912.1 words for GovReport-sourced QA. The question types are also broadly distributed: the most frequent type, *Multi-Issue Synthesis*, accounts for 17 instances, while the remaining six types range from 10 to 15 instances. Across all QA instances, the 480 referenced rules cover all six semantic augmentation types: 86 **ABSTRACT**, 82 **ADDITIVE**, 75 **NEGATE**, 82 **EXCEPTION**, 89 **CONFLICT**, and 66 **IRONCLAD** rule mentions.

B Additional Details of RuleWeaver

B.1 Representative Semantic Augmentation Examples

Table 9 presents representative examples selected from the organized final rule set. For readability, the table reports the source Meta Rule and the specific augmented information that carries the semantic type, rather than printing the full five-round complex rule. These examples illustrate how the

Type	Source Meta Rule	Augmented Information	Augmentation Definition
ABSTRACT	If a third-party claim arises from an intellectual property infringement, then the indemnifying party should defend the indemnified party.	The condition is expanded with the broader alternative "or any other similar legal wrong."	This abstracts the concrete trigger, *intellectual property infringement*, into a higher-level legal category, requiring models to match the rule beyond surface lexical overlap.
ADDITIVE	If an item is controlled on the applicable export control list, then the exporter should submit a license application to the relevant regulatory authority.	Added condition: "the item is not explicitly pre-approved for export under a special government-to-government agreement." Added outcome: "not proceed with the export without first obtaining a binding advisory opinion."	The new information does not reverse or suspend the export-control rule; it supplements the original scope with an additional eligibility condition and an additional compliance requirement.
NEGATE	If a site scores above 28.5 on the applicable hazard assessment metric, then the site is eligible for placement on the relevant national priority list.	The rule becomes: if the site "does not score above 28.5," then it is "not eligible for placement."	Both the IF condition and THEN outcome are reversed, testing whether a model tracks the changed semantic direction instead of applying the original eligibility rule.
EXCEPTION	If a party wishes to assign the Agreement, then that party should obtain prior written consent from the other party.	Added exception: "except where the assignment is to an affiliate."	The added affiliate case restricts the general consent requirement by specifying a special circumstance in which the rule should not be directly applied.
CONFLICT	If the service provider uses commercially reasonable efforts to correct a Program Error, then failure to correct it within the specified period shall not be deemed a breach.	Added competing condition: if the failure results from "willful misconduct or fraud," then the failure "shall be deemed a breach" and triggers penalties.	The added rule creates an incompatible outcome with the original no-breach conclusion, requiring conflict detection and resolution when both conditions are relevant.
IRONCLAD	If a federal agency intends to dispose of property, then the agency should assess the need for environmental cleanup.	Added mandatory prohibition: the agency "must not transfer the property to any non-federal entity before the assessment is complete."	The new item is framed as a strict, non-optional constraint that must be preserved even when other outcomes, waivers, or procedural alternatives are considered.

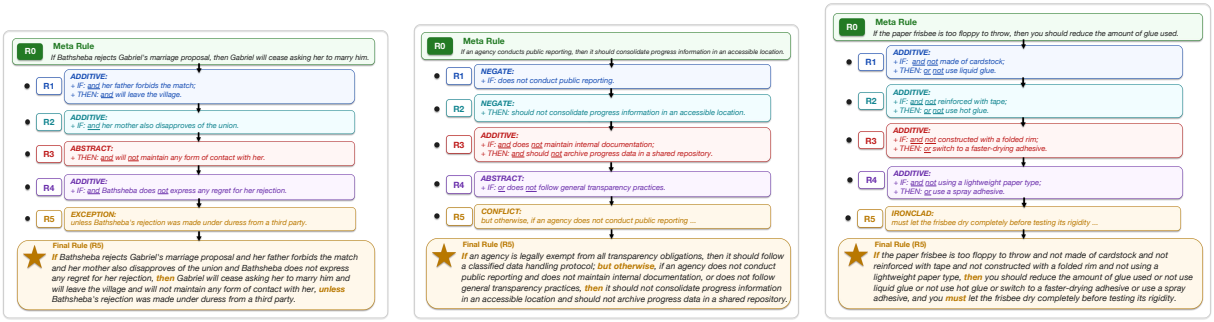
Table 9: Representative examples of the six semantic augmentation types selected from the organized final rule set.

same IF-THEN backbone can be enriched in different semantic directions while preserving an explicit record of the augmentation operator and its intended reasoning effect. The selected examples emphasize that the six augmentation types differ not only in surface wording, but also in the reasoning behavior they induce. **ABSTRACT** changes the granularity of a rule term, **ADDITIVE** integrates further requirements within the same rule scope, and **NEGATE** changes the direction of rule applicability or consequence. The higher-impact types operate over rule applicability and priority: **EXCEPTION** carves out a special non-applicable case, **CONFLICT** introduces a competing rule with an incompatible conclusion, and **IRONCLAD** adds a constraint that should remain active despite other rule interactions.

The table above illustrates isolated, single-step semantic augmentation events. To complement these examples, Figure 4 shows three complete five-round progressive augmentation cases. Each case preserves its full augmentation trajectory from the source Meta Rule to the final complex rule: the final round realizes an **EXCEPTION** in Figure 4a, a **CONFLICT** in Figure 4b, and an **IRONCLAD** constraint in Figure 4c. These cases make explicit how earlier moderate augmentations are accumulated before a high-impact rule interaction is introduced in the final round.

Source	Meta Rule	IF Condition	THEN Outcome
BookSum	If a person is excommunicated from the Church, then they are banished from entering Rome.	A person is excommunicated from the Church	They are banished from entering Rome
CUAD	If the manufacturer intends to change a manufacturing process, then the manufacturer should obtain written approval from the client via a formal change order.	The manufacturer intends to change a manufacturing process	The manufacturer should obtain written approval from the client via a formal change order
GovReport	If the postal service provider proposes to eliminate a standard delivery day, then the provider should request an advisory opinion from the relevant regulatory oversight body.	The postal service provider proposes to eliminate a standard delivery day	The provider should request an advisory opinion from the relevant regulatory oversight body
WikiHow	If a tank lacks a bubbler, then the water becomes acidic.	A tank lacks a bubbler	The water becomes acidic

Table 10: Examples of Meta Rules from Four Source Datasets.



(a) **EXCEPTION** (b) **CONFLICT** (c) **IRONCLAD**

Figure 4: Complete five-round progressive augmentation cases, with final semantic enhancements of **EXCEPTION**, **CONFLICT**, and **IRONCLAD**, respectively.

B.2 Implementation Details for Complex Rule Construction

For representative sampling, documents are embedded with Qwen3-8B-Embedding (Zhang et al., 2025) and clustered with KMeans (MacQueen, 1967), using 50 clusters per dataset. We select the 3 nearest documents from each cluster, split them into chunks of up to 3,000 tokens with CL100K_Base in tiktoken (OpenAI, 2026c), and generate 10 Meta Rules per chunk with GPT-5.4 (OpenAI, 2026b) using the prompt in Figure 18. For quality filtering, malformed outputs are removed before Grok-3-Mini (xAI, 2025) verifies rule atomicity with the prompt in Figure 19. We then remove rules that are under-informative, overly commonsensical, weak in constraint semantics, or dependent on external background knowledge through unanimous voting by Grok-3-Mini, GPT-4.1-Mini (OpenAI, 2025), and Gemini-3.1-Flash-Lite-Preview (Google AI Blog, 2026), using the prompt in Figure 20. The retained rules are embedded with Qwen3-8B-Embedding and clustered separately for each dataset using KMeans with 50 clusters per dataset; the most representative rule from each cluster is then manually inspected. Progressive complex rule construction uses DeepSeek-V4-Pro (DeepSeek-AI et al., 2026) with the rule augmentation prompt in Figure 21.

B.3 Representative Meta Rule Examples

Table 10 shows representative Meta Rules extracted from the four source datasets used in RuleWeaver. Each example follows the atomic IF-THEN format, with one triggering condition and one corresponding outcome. Together, these examples illustrate the source diversity of the seed rules before semantic augmentation: policy-oriented obligations from GovReport, procedural guidance from WikiHow, contractual requirements from CUAD, and event-contingent rules from BookSum.

B.4 Scenario QA Question Types

RuleWeaver samples one target question type before constructing the dependency plan for each scenario QA instance. The type controls the intended reasoning demand of the final question, while the hidden dependency plan and rule logic chain en-

Question Type	Expected Output	Reasoning Focus
Multi-Issue Synthesis	A synthesis of multiple parallel and relevant points that jointly resolve the question.	Integrating several sub-issues instead of producing a single isolated conclusion.
Definitive Conclusion	One unique, explicit conclusion derived from the scenario, question, and applicable rules.	Checking all relevant rule conditions and deriving a determinate final answer.
Event Prediction	The rule-governed next event or consequence if the current scenario proceeds.	Projecting consequences beyond the immediately stated facts while remaining grounded in the rules.
Priority Arbitration	The controlling rule-governed path and the resulting choice or conclusion.	Resolving rule interactions involving conflicts, competing implications, or priority ordering.
Special-Case Judgment	The outcome for a boundary case or special circumstance.	Determining how a qualifying circumstance changes rule applicability or the final outcome.
What-If Revision	The revised rule-governed outcome after a specified change to scenario conditions.	Updating the reasoning chain under a counterfactual modification rather than reusing the original conclusion.
Action Prescription	The concrete action, disposition, or set of actions that should be taken next.	Translating rule applications and dependencies into an operational decision.

Table 11: Scenario QA question types used in RuleWeaver.

sure that the instance still requires grounded rule application rather than surface pattern matching. Table 11 summarizes the seven question types used in RuleWeaver.

B.5 Scenario QA Rubric Dimensions

The answer judge scores each model response on a 100-point rubric with nine dimensions. The rubric separates surface validity, such as citation format and avoiding external facts, from rule-centered reasoning skills, such as decomposition, dependency alignment, and exception or conflict handling. Table 12 summarizes the dimensions and their maximum points.

B.6 Human Agreement Audit for Rubric Scoring

To assess the reliability of the LLM-based answer judge, we conduct a score-stratified audit on 48 GPT-5.5 answers from the 200-rule setting, sampling 12 answers from each of four automatic rubric-score bins: $[0, 25)$, $[25, 50)$, $[50, 75)$, and $[75, 100]$. One human annotator independently rescores the answers using the same instance-specific all-or-nothing rubric, with automatic scores hidden during annotation.

We further extend the audit to outputs from four representative models: Claude-Opus-4.6, Claude-Sonnet-4.6, DeepSeek-V4-Pro, and GPT-5.4. For each model, we rank its outputs by automatic rubric score, partition the ranked outputs into four percentile ranges (top 25%, 25–50%, 50–75%, and bottom 25%), and randomly select 12 responses from each range. Two human annotators independently rescore the same 48 selected responses per model with automatic scores hidden. This percentile-stratified design covers low- and high-scoring outputs even when a model does not populate the fixed score intervals uniformly.

C Additional Evaluation Results

C.1 Bootstrap Confidence Intervals

For each source setting, we resample the 48 aligned QA instances with replacement 20,000 times and recompute the mean rubric score. We use the 2.5th and 97.5th percentiles of the bootstrap distribution to form 95% confidence intervals (CIs). Table 13 reports the resulting model-level estimates. The bootstrap means differ from the corresponding point estimates in Table 2 by at most 0.43 points. We additionally compare the leading model in each setting with every other model using a paired bootstrap over the same 48 aligned QA instances. Table 14 reports the mean difference and 95% CI for each comparison.

C.2 Semantic Enhancement Results under the 200-Rule Setting

Table 15 reports model performance across the six semantic enhancement types under the 200-rule setting. We use the same event-level semantic correct-application rate as in the main radar plot: for a semantic type, the denominator is the set of corresponding rule-enhancement events attached to the rules referenced by evaluated questions, and the numerator counts events whose rule is both cited and judged as correctly applied in the model answer. Higher values indicate stronger semantic rule-following ability. The results show that Claude-

Rubric Dimension	Max	What It Measures
Question Understanding	10	Whether the answer addresses the requested scenario question and identifies the correct target outcome or decision.
Issue Decomposition	10	Whether the answer breaks the scenario into the necessary sub-issues and rule-application steps before reaching the final conclusion.
Citation Format Compliance	8	Whether cited rules use canonical rule identifiers, appear at the point of application, and avoid invented or malformed rule references.
Rule-Grounded Reasoning	17	Whether cited rules are correctly applied to concrete scenario facts and support the stated intermediate or final conclusions.
Dependency Chain Alignment	18	Whether the answer follows the required *rule logic chain* and preserves dependencies among intermediate conclusions.
Exception/Conflict Handling	12	Whether the answer detects and resolves exceptions, blockers, conflicts, or priority relations that affect rule applicability.
Intermediate Conclusion Quality	10	Whether the answer derives accurate intermediate conclusions for the necessary rules instead of jumping directly to the final answer.
Final Answer Consistency	12	Whether the final answer follows from the preceding rule-grounded reasoning and states the required conclusion, action, or judgment.
No External Facts	3	Whether the answer avoids unsupported assumptions or facts outside the provided scenario and rule pool.

Table 12: Scenario QA answer-judgment rubric dimensions. The maximum scores sum to 100 points.

	Same-Source			Cross-Source		
Model	Mean	95% CI	Δ	Mean	95% CI	Δ
GPT-5.5	53.88	[44.42, 63.33]	+0.05	46.49	[38.21, 54.77]	+0.09
Claude-Opus-4.6	48.21	[38.65, 57.77]	+0.15	50.43	[40.81, 60.04]	+0.16
GPT-5.4	48.39	[38.40, 58.38]	+0.18	42.67	[33.83, 51.50]	+0.09
Claude-Sonnet-4.6	44.11	[33.90, 54.31]	+0.15	36.69	[27.38, 46.00]	+0.13
Qwen3.5-Plus	38.03	[29.00, 47.06]	+0.28	34.47	[26.65, 42.29]	+0.32
GLM-5	37.65	[28.77, 46.52]	+0.11	32.66	[25.85, 39.46]	+0.31
DeepSeek-V4-Pro	34.95	[25.27, 44.63]	+0.26	28.75	[22.52, 34.98]	+0.37
Doubao-Seed-2.0-Pro	28.03	[19.40, 36.65]	+0.36	33.35	[26.00, 40.69]	+0.33
Kimi-K2.6	32.88	[24.23, 41.52]	+0.38	28.90	[19.96, 37.83]	+0.42
Gemini-3.1-Pro-Preview	32.12	[23.29, 40.94]	+0.35	28.23	[20.96, 35.50]	+0.42
MiniMax-M2.7	30.07	[20.38, 39.75]	+0.38	21.58	[13.75, 29.40]	+0.43

Table 13: Bootstrap mean rubric scores and 95% CIs. Δ is the bootstrap mean minus the corresponding point estimate in Table 2.

Opus-4.6 is the strongest overall model, achieving the best score on **ABSTRACT**, **ADDITIVE**, **NEGATE**, and **CONFLICT**, and tying GPT-5.5 on **IRONCLAD**. Claude-Sonnet-4.6 performs best on **EXCEPTION**, suggesting comparatively strong handling of special-case carve-outs. Aggregated across models, **CONFLICT** and **IRONCLAD** are the highest-scoring semantic types, while **EXCEPTION**, **NEGATE**, and **ADDITIVE** remain more challenging, indicating that models still struggle with precise applicability changes and condition-level modifications.

C.3 Question-Type Results under the 200-Rule Setting

Table 16 reports the model-level rubric scores for the seven scenario question types under the 200-rule setting. The question types are defined in Appendix B.4; higher scores indicate better performance.

C.4 Model-Level Rubric-Dimension Breakdown

Figure 5 breaks down normalized rubric scores by model. The same pattern is broadly shared across model families: even stronger models improve on rule-grounded reasoning and final-answer consistency, but dependency-chain alignment remains low. These results complement the aggregate rubric scores by localizing where models lose points in the reasoning process.

C.5 QA-Level Spearman Correlation Analysis

We complement the descriptive scenario-complexity analysis in Section 4.4 with QA-level rank correlations. For each structural or rule-selection metric, let $\mathbf{x} = (x_1, \ldots, x_N)$ be its values over the N QA instances, and let $\mathbf{y} = (y_1, \ldots, y_N)$ be the corresponding QA-level rubric scores averaged over all evaluated models. We report Spearman's rank correlation coefficient,

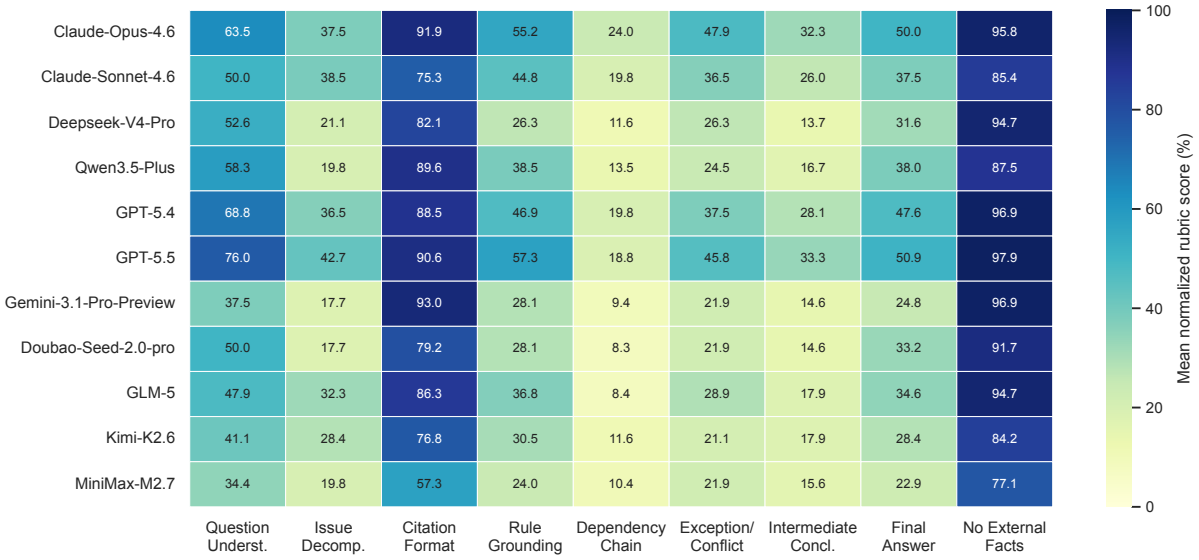


Figure 5: Model-level normalized rubric-dimension scores under the 200-rule setting. Each cell reports the mean percentage score for one model and one rubric dimension.

Same-source comparison	Diff.	95% CI
GPT-5.5 – Doubao-Seed-2.0-Pro	26.36	[16.62, 36.10]
GPT-5.5 – MiniMax-M2.7	24.08	[12.75, 35.40]
GPT-5.5 – Gemini-3.1-Pro-Preview	21.96	[11.15, 32.77]
GPT-5.5 – Kimi-K2.6	21.46	[10.81, 32.10]
GPT-5.5 – DeepSeek-V4-Pro	19.19	[8.38, 30.00]
GPT-5.5 – GLM-5	16.48	[7.42, 25.54]
GPT-5.5 – Qwen3.5-Plus	16.27	[6.00, 26.54]
GPT-5.5 – Claude-Sonnet-4.6	9.87	[0.92, 18.81]
GPT-5.5 – GPT-5.4	5.72	[-3.62, 15.06]
GPT-5.5 – Claude-Opus-4.6	5.60	[-5.04, 16.23]

Cross-source comparison	Diff.	95% CI
Claude-Opus-4.6 – MiniMax-M2.7	29.30	[18.62, 39.98]
Claude-Opus-4.6 – Gemini-3.1-Pro-Preview	22.77	[13.00, 32.54]
Claude-Opus-4.6 – DeepSeek-V4-Pro	22.02	[11.46, 32.58]
Claude-Opus-4.6 – Kimi-K2.6	21.99	[11.62, 32.35]
Claude-Opus-4.6 – GLM-5	18.03	[7.83, 28.23]
Claude-Opus-4.6 – Doubao-Seed-2.0-Pro	17.36	[7.50, 27.21]
Claude-Opus-4.6 – Qwen3.5-Plus	16.17	[5.67, 26.67]
Claude-Opus-4.6 – Claude-Sonnet-4.6	13.93	[3.71, 24.15]
Claude-Opus-4.6 – GPT-5.4	7.71	[-1.23, 16.65]
Claude-Opus-4.6 – GPT-5.5	4.13	[-3.92, 12.17]

Table 14: Paired bootstrap rubric-score differences (leader minus comparator) and 95% CIs, based on 20,000 resamples of the 48 aligned instances in each setting.

$\rho = \mathrm{corr}(\mathrm{rank}(\mathbf{x}), \mathrm{rank}(\mathbf{y}))$, where tied values receive their average rank.

Table 17 shows modest negative correlations for reasoning depth ($\rho = -0.29$) and dependency load ($\rho = -0.22$), while structural branching is nearly uncorrelated with score ($\rho = -0.01$). For rule-selection factors, rule precision is strongly associated with final score ($\rho = 0.66$), whereas rule recall is weaker ($\rho = 0.20$). This indicates that correctly applying cited rules is more directly tied to answer quality than merely citing more required rules.

C.6 Case Studies

To better understand the limitations of current LLMs in compositional rule-following, we present several representative failure cases. Since the full scenarios, rule contents, and model outputs are lengthy, we report only the case-relevant core information. These cases show errors such as missing individual rules, misusing exception conditions, and failing to propagate intermediate conclusions across dependent rules.

LLMs fail to recall necessary rules. As illustrated in Figure 6, LLMs may fail to retrieve a rule that is essential for completing the reasoning chain. In this case, the model applies several local exposure rules but omits R(117), the rule that aggregates the resulting plan states into concrete premium and corrective-funding obligations. As a result, the model leaves the obligations unresolved and treats them as pending or contingent, rather than deriving the determinate duties required by the rule set.

LLMs misuse exception conditions. As shown in Figure 7, LLMs may identify the relevant facts but reverse the polarity of an exception. In this case, unpaid fees accrue interest unless an effective written forbearance agreement exists. The scenario

Model	ABSTRACT	ADDITIVE	NEGATE	EXCEPTION	CONFLICT	IRONCLAD
Claude-Opus-4.6	**50.55**	**48.02**	**46.08**	40.24	**56.18**	**51.52**
Claude-Sonnet-4.6	39.45	38.24	39.06	**42.68**	41.57	39.39
Deepseek-V4-Pro	37.53	32.58	31.50	36.59	41.57	37.88
Qwen3.5-Plus	41.23	37.25	34.39	32.93	40.45	42.42
GPT-5.4	44.11	41.93	40.85	36.59	49.44	46.97
GPT-5.5	46.99	43.34	45.12	40.24	51.69	**51.52**
Gemini-3.1-Pro-Preview	39.04	35.98	33.98	34.15	43.82	40.91
Doubao-Seed-2.0-pro	32.33	29.60	28.89	24.39	40.45	28.79
GLM-5	38.77	34.42	35.35	35.37	34.83	40.91
Kimi-K2.6	37.95	36.97	33.84	31.71	37.08	40.91
MiniMax-M2.7	28.49	24.93	23.93	25.61	32.58	25.76

Table 15: Semantic correct-application rates by enhancement type under the 200-rule setting. Scores are percentages, and bold values mark the best model for each semantic type.

Model	Multi-Issue Synthesis	Definitive Conclusion	Event Prediction	Priority Arbitration	Special-Case Judgment	What-If Revision	Action Prescription
Claude-Opus-4.6	**45.24**	**55.64**	45.00	**52.43**	51.73	50.17	43.00
Claude-Sonnet-4.6	39.18	40.14	40.21	24.79	48.53	44.33	**46.70**
GPT-5.4	38.94	53.36	48.93	42.79	53.33	47.17	29.90
GPT-5.5	40.94	54.86	**57.50**	38.79	**60.73**	**52.67**	45.60
Gemini-3.1-Pro-Preview	27.18	52.64	33.79	13.21	30.07	26.50	23.40
Deepseek-V4-Pro	30.06	39.79	34.07	24.64	27.93	35.17	29.60
Qwen3.5-Plus	40.24	48.14	37.21	26.79	32.73	37.08	26.10
GLM-5	21.24	40.64	30.14	25.43	57.00	37.50	34.20
Doubao-Seed-2.0-pro	28.88	34.36	28.79	23.07	31.87	36.50	29.90
MiniMax-M2.7	23.06	26.64	26.07	22.07	34.93	24.75	18.00
Kimi-K2.6	30.94	31.36	14.64	18.71	49.27	43.83	23.00

Table 16: Rubric scores by scenario question type under the 200-rule setting. Scores are on a 0–100 scale, and bold values mark the best model for each question type.

Group	Metric	Spearman ρ
Complexity	Reasoning Depth	-0.29
	Dependency Load	-0.22
	Branching	-0.01
Rule Selection	Rule Recall	0.20
	Rule Precision	0.66

Table 17: Spearman rank correlations between QA-level factors and mean rubric score. Rule recall and precision are averaged over the 11 evaluated models for each QA.

only contains an email promising a future draft, so the exception should not apply. However, the model interprets the absence of a written forbearance agreement as blocking interest accrual. This reverses the rule consequence and produces a no-interest outcome.

LLMs lose dependency-chain conclusions. As shown in Figure 8, LLMs may apply individual rules locally while failing to propagate intermediate conclusions across dependent rules. In this case, R(83) establishes that the corrected disclosure blocks notification or escalation. This conclusion should then inform the later invention-documentation and amendment analyses under R(54) and R(68). However, the model treats R(83) as an isolated resolved item and evaluates the later rules separately. Consequently, locally plausible rule applications fail to produce the required global reasoning chain.

Figures 9–17 additionally present representative scoring cases for all nine rubric dimensions. Each case shows a model response that receives zero points on the corresponding dimension, together with the relevant scenario evidence, reference answer, and judge rationale.

Case Study: Recall Missing

Scenario: ... The sponsor faces several local exposures: a denied cost-relief application, a prohibited twisting lift, an uncured remarketing failure under a regulatory stay, and infernal-origin artifact damage. These exposures collectively determine whether the pension plan is underfunded, non-compliant, or high-risk. ...

Gold Rules (partial rule contents): R(180) denies cost relief for a vexatious litigant; R(160) prohibits twisting lifts for a spinal-injury worker; R(52) lifts tender-right suspension after an uncured failed remarketing event with a regulatory stay; R(50) establishes the infernal-origin harm; R(117) converts plan underfunding / non-compliance / high-risk status into premium and corrective-funding obligations.

Question: What obligations and financial exposures must the pension-plan sponsor address?

Model Outputs: ... Plan status remains unresolved; no definitive variable-rate premium or corrective-funding obligation is currently required. The sponsor only needs to prepare for pending premium and corrective-proposal deadlines, while the lawsuit, warehouse incident, remarketing event, and artifact damage remain separate exposures. ...

Correct: ... The sponsor must pay the variable-rate premium, fulfill the alternative financial obligation, and submit the corrective funding proposal under R(117), while also absorbing the lawsuit, warehouse, remarketing, and artifact-loss exposures. ...

Figure 6: **Recall missing.** The model misses a necessary aggregation rule and therefore treats determinate obligations as merely contingent. Red text highlights the erroneous part of the model output.

Case Study: Rule Misuse

Scenario: ... BlueLine has $2,000 in unpaid fees. The contract charges 1.5% monthly interest unless a written forbearance agreement is in effect. The only evidence is an email promising a future draft, so no effective written forbearance exists. ...

Gold Rules (partial rule contents): R(92) controls the interest consequence; when no written forbearance agreement is effective, unpaid fees accrue 1.5% monthly interest. The other rules provide surrounding regulatory context but do not change the final payment calculation.

Question: What is BlueLine Logistics' final aggregated financial obligation to FastFunds as of April 1?

Model Outputs: ... BlueLine owes $2,000 in unpaid fees. No written forbearance agreement is in effect. Therefore, the unpaid amount shall not bear 1.5% monthly interest, and the final obligation is $2,000. ...

Correct: ... The final obligation is $2,030: $2,000 unpaid principal plus $30 interest at 1.5% for one month. Because no written forbearance agreement is in effect, the exception does not block interest. ...

Figure 7: **Rule misuse.** The model identifies the relevant premise but reverses the exception condition, producing the opposite legal consequence. Red text highlights the erroneous part of the model output.

Case Study: Dependency Incorrect

Scenario: ... Helion has four open compliance items. The key linking fact is that the material misstatement was fully corrected and publicly disclosed before the designated agent sent notice. That corrected-disclosure conclusion should guide the later invention-documentation and amendment analyses. ...

Gold Rules (partial rule contents): R(51) removes the need for a reseller agreement; R(70) bars cost pass-through for a purely in-house service; R(83) blocks notification/escalation after pre-notice correction; R(54) governs invention assignment, patent filing, and secure logs; R(68) requires an executed written/electronic amendment for material-term changes.

Question: What is Helion Ventures' overall compliance posture as of March 25, 2025?

Model Outputs: ... The reseller and cost items are compliant. The corrected disclosure requires no notification or escalation. The NovaTech invention records and Apex payment amendment are then treated as separate documentation and amendment risks, so the overall posture is mixed. ...

Correct: ... R(83)'s corrected-disclosure conclusion must feed into R(54) and R(68). Helion need not execute a reseller contract; CloudCore must absorb costs; Helion must secure invention logs without assignments or patent filings; and the unsigned payment amendment is not effective. ...

Figure 8: **Dependency incorrect.** The model applies rules locally but fails to propagate an intermediate conclusion through the required rule-dependency chain. Red text highlights the erroneous part of the model output.

Failure Case: Question Understanding

Question Understanding · 0/10

Key scenario excerpt: "A traveling tinker, who had been given shelter in the barn the previous night, had been found shouting at the milkmaids and overturning pails. ... The tinker, a man named Jack, had been warned just last week by the farm steward for drunkenness and for letting his dog chase the sheep."
Question: If the lamb had not been surrendered, what would happen to Jack after his disruption?
Key rule R(33): A disruptive guest with problematic behavior and a prior formal warning should not be removed or warned again.

Doubao-Seed-2.0-Pro answer
"Rule R(30) ... applies to a farm worker caught thieving. Final Answer: dismiss Jack and issue him a formal final warning."

Reference Answer
"Under supplied R(33), the host would still not remove Jack and would not issue a formal warning."

Judge rationale (DeepSeek-V4-Flash)
The answer incorrectly interprets the question as asking for the outcome under Rule R(30) about farm worker theft, whereas the final question is about the outcome for Jack after his disruption under the counterfactual that the lamb was not surrendered. The tested answer fails to identify the exact outcome requested.

Figure 9: A zero-score case for *Question Understanding*.

Failure Case: Issue Decomposition

Issue Decomposition · 0/10

Key scenario excerpt: "The file was a standard Word document, not a PDF/X-1a or packaged InDesign file. ... Beta sent Acme a revised labeling file in PDF/X-1a format ... Acme had not sent a formal notice about the labeling format back in March ... the snowstorm had indeed closed Beta's office for three days"
Question: What action is required of Acme Corp under the agreement with Beta Supplies?
Required subquestions: termination/suspension; timely cure; force majeure; substantial deprivation; written waiver.

Gemini-3.1-Pro-Preview answer
"Final Answer: Acme should not provide artwork, notify Beta, suggest conversion, or propose another resolution under R(98)."

Reference Answer
"The required disposition is to continue performance, accept the late remedy, and refrain from termination or suspension under R(55)."

Judge rationale (DeepSeek-V4-Flash)
The answer does not decompose the final question into sub-issues such as termination under R(55), force majeure, substantial deprivation of benefits, or waiver; it only addresses R(96) without connecting to other sub-issues.

Figure 10: A zero-score case for *Issue Decomposition*.

Failure Case: Citation Format Compliance

Citation Format Compliance · 0/8

Key scenario excerpt: "Jim shoved him hard against the pole, then turned on Dorian ... Angel, after a night of turmoil, forgave her ... as he struck the canvas, he himself collapsed, dead, his face withered while the portrait returned to its original beauty."
Question: Compare the fates of Jim, the d'Urbervilles, and Dorian Gray, then explain the combined outcome.
Provided rule IDs: R(19), R(14), R(29), R(13), and R(36).

Kimi-K2.6 answer
"R(37)'s exception for those 'marked by a curse of reflection' ensures that destroying the portrait kills Dorian physically."

Reference Answer
"The integrated outcome must be derived from the five supplied rules; R(37) cannot support any conclusion."

Judge rationale (DeepSeek-V4-Flash)
The answer uses canonical rule citations (e.g., R(36), R(29), R(13)) for the relevant rules, but also cites R(37) which is not among the provided relevant rules. The reference answer context lists only rules 13, 14, 19, 29, 36 as relevant. R(37) does not appear anywhere in the provided rules or reference context, so it is an invented rule ID.

Figure 11: A zero-score case for *Citation Format Compliance*.

Failure Case: Rule-Grounded Reasoning

Rule-Grounded Reasoning · 0/17

Key scenario excerpt: "OmniCorp never formally assumed NovaTech's obligations under the agreement. ... Tom drafted an assignment request ... she could not consider consent until the indemnification matter involving the service provider was clarified ... the provision affecting the GlobalSync offer was formally resolved."
Question: What termination right does Apex have, and may ArtStream assign MelodyVault before resolving legacy issues?
Key rules: R(56) permits termination after notice; R(67) requires prior written consent before assignment.

Doubao-Seed-2.0-Pro answer
"No rule in the provided set addresses either scenario. Final Answer: there is insufficient relevant information."

Reference Answer
"Apex may terminate by notice under R(56); ArtStream's assignment remains blocked until the listed issues are resolved."

Judge rationale (DeepSeek-V4-Flash)
The tested answer does not apply any cited rules to scenario facts; it claims no rules govern the scenarios, thus failing to provide rule-grounded reasoning.

Figure 12: A zero-score case for *Rule-Grounded Reasoning*.

Failure Case: Dependency-Chain Alignment

Dependency Chain Alignment · 0/18

Key scenario excerpt: "The file was a standard Word document, not a PDF/X-1a or packaged InDesign file. ... Tom sent Beta a formal notice of material breach ... Beta sent Acme a revised labeling file in PDF/X-1a format ... any termination attempt would likely fail given the contractual provisions and the lack of clear evidence."

Question: What action is required of Acme Corp under the agreement with Beta Supplies?
Required chain: conclusions from R(96), R(65), R(76), and R(87) feed into the R(55) termination analysis.

Gemini-3.1-Pro-Preview answer

"Final Answer: Acme should not provide artwork, notify Beta, suggest conversion, or propose another resolution under R(96)."

Reference Answer

"The required disposition is to continue performance, accept the late remedy, and refrain from termination or suspension under R(55)."

Judge rationale (DeepSeek-V4-Flash)

The answer does not follow the dependency chain; it treats R(96) as an independent conclusion without integrating prior conclusions from R(96), R(65), R(76), R(87) into an R(55) analysis.

Figure 13: A zero-score case for *Dependency-Chain Alignment*.

Failure Case: Exception/Conflict Handling

Exception/Conflict Handling · 0/12

Key scenario excerpt: "Over six months had passed with no activity and no written explanation, despite an unpaid freight invoice ... the fertilizer had been used in a greenhouse with automated misting that exceeded the product's moisture limits ... the customer had skipped the recommended pre-use inspection."

Question: What is FreshFields' final legal position across the purchase, review, warranty, and settlement issues?
Key rule: R(62) requires no misuse or abnormal conditions and completion of the mandatory pre-use inspection.

GLM-5 answer

"Since the factual conditions triggering cessation are met ... FreshFields' warranty obligations cease."

Reference Answer

"The R(62) antecedent is not met; warranty obligations do not cease and FreshFields remains liable."

Judge rationale (DeepSeek-V4-Flash)

The answer incorrectly handles R(62) exceptions: it treats customer misuse as triggering cessation, but fails to recognize that R(62) requires absence of misuse, abnormal conditions, etc.; also ignores outstanding settlement as blocker.

Figure 14: A zero-score case for *Exception/Conflict Handling*.

Failure Case: Intermediate Conclusions

Intermediate Conclusion Quality · 0/10

Key scenario excerpt: "The room had a microwave, a ceiling fan, and a window that didn't latch properly. ... The vet had said Kiwi needed a follow-up, but Mara hadn't rescheduled. ... Mara let Kiwi out again, thinking the bird needed exercise, and didn't tell anyone in the adjacent cubicles that Kiwi was loose."

Question: What is the overall outcome for Kiwi's safety and Mara's actions?
R(155) conclusion: Allow Kiwi to remain loose and unsupervised, without monitoring, and do not notify household members.

Claude-Sonnet-4.6 answer

"Mara was obligated to secure Kiwi, supervise the bird, and inform others. Mara violated all of these obligations."

Reference Answer

"The answer reverses every operative conclusion in R(155); Mara's actions actually match the supplied rule."

Judge rationale (DeepSeek-V4-Flash)

The intermediate conclusion about R(155) is incorrect (reverses the rule's prescription), and no other intermediate conclusions from the decomposition are present.

Figure 15: A zero-score case for *Intermediate Conclusions*.

Failure Case: Final Answer Consistency

Final Answer Consistency · 0/12

Key scenario excerpt: "an investigation showed the fertilizer had been used in a greenhouse with automated misting that exceeded the product's moisture limits, and the customer had skipped the recommended pre-use inspection. ... FreshFields insisted the mold was a manufacturing defect and demanded warranty coverage."

Question: What is FreshFields' final legal position across the purchase, review, warranty, and settlement issues?
Correct intermediate result: Misuse, abnormal conditions, and the skipped inspection defeat R(62), so the warranty does not cease.

GLM-5 answer

"Since the factual conditions triggering cessation are met ... FreshFields' warranty obligations cease."

Reference Answer

"The R(62) antecedent is not met; warranty obligations do not cease and FreshFields remains liable."

Judge rationale (DeepSeek-V4-Flash)

Final answer states warranty obligations cease, directly contradicting the reference final answer; thus does not follow from correct intermediate conclusions.

Figure 16: A zero-score case for *Final Answer Consistency*.

Failure Case: Unsupported External Facts

No External Facts · 0/3

Key scenario excerpt: "a protected spreadsheet where a cell range for counseling referrals is unlocked, visible, and free of formatting restrictions, but currently has the default white fill. ... apply a soft yellow, light green, or another noticeable but non-alarming color, avoiding shades reserved for error flags."

Question: What fill color should be applied to the counseling-referrals input range?
Key rule: R(151) permits canary yellow, light green, or another distinguishing color that is not an error indicator.

GLM-5 answer

"Light green is universally associated with positive, safe, and managed states ... Final Answer: Light green."

Reference Answer

"The preference for light green is supported only by an external value judgment, not by supplied evidence."

Judge rationale (DeepSeek-V4-Flash)

The answer introduces an unsupported external fact: 'Light green is universally associated with positive, safe, and managed states.' This is a value judgment not provided in the scenario or rules, violating the rule.

Figure 17: A zero-score case for *No External Facts*.

Meta Rule Extraction Prompt

Please extract or summarize {max_num} Meta Rules from the given text passage and output them as a JSON list.

Requirements

1. A Meta Rule should be expressed as an explicit "if ..., then ..." statement.

2. Each rule must contain exactly one IF condition and one THEN outcome.
 - The IF field must contain exactly one indivisible triggering proposition.
 - The THEN field must contain exactly one indivisible outcome or requirement.
 - Do not use "and", "or", "and/or", "whether ... or ...", or similar coordinating expressions within either the IF side or the THEN side. Such structures should be split into separate rules whenever possible.

3. The generated rules must have high semantic diversity and be clearly distinguishable in meaning.
 - Do not produce paraphrases of the same rule.
 - Do not produce near duplicates that differ only in wording, tense, or surface phrasing.
 - Each rule should capture a meaningfully different regulatory, conditional, or obligation pattern supported by the text.

Output Format

Output only a valid JSON list.
Do not nclude any extra explanation, headings, or Markdown code fences.

Each element must use the following structure:

```
{
 "rule": "...",
 "IF": "...",
 "THEN": "...",
 "rule_explanation": "...",
 "rule_in_chinese": "..."
}
```

Context

{context}

Figure 18: Meta rule extraction prompt.

IF-THEN Filtering Prompt

You are a strict quality assurance reviewer for regulatory rules. Your task is to evaluate a rule's structure and filter it out if it is not strictly atomic.

Evaluation Criteria

A valid rule must strictly contain exactly one single condition in the "IF" clause, and exactly one single action or outcome in the "THEN" clause. You must filter out the rule (set `should_filter` to true) if the IF side contains multiple conditions OR if the THEN side contains multiple actions/results.

Rule to evaluate

{rule}

Output Format (respond with only JSON, no other text)

```
{
 "should_filter": true/false,
 "reason": "Brief explanation if filtered out , or null if kept"
}
```

Figure 19: IF-THEN filtering prompt.

Meta Rule Quality Filtering Prompt

You are an expert data curator assessing a dataset designed to evaluate the rule-based logical reasoning capabilities of large language models.

Your task is to determine whether a given rule should be excluded from the dataset according to the following exclusion criteria:

1. Low Informational Value:
The rule is trivial, redundant, circular, underspecified, or otherwise lacks sufficient substantive content.
{rule_example_low_info}

2. Dependence on External or Unprovided Legal Acts:
The rule depends on highly specific legal provisions, statutes, regulations, or legal acts that are not provided in the dataset. Note that the mere presence of a specific entity name does not, by itself, warrant exclusion.
{rule_example_external_or_unprovided_legal_act}

3. Commonsense-Based Reasoning:
The rule relies primarily on ordinary commonsense knowledge or highly intuitive reasoning, and therefore provides limited additional value for evaluating formal rule-based reasoning.
{rule_example_common_sense}

Input Rule:
{rule}

Evaluate whether the rule should be excluded based on the criteria above.

Output Format:
Respond only with a valid JSON object in the following format:
{
"should_filter": true/false,
"reason": "A concise explanation if the rule is excluded, or null if it is retained."
}

Figure 20: Meta rule quality filtering prompt.

Rule Augmentation Prompt

Universal Template:

You are given a root rule.

Rule: {rule}

Your task is to generate semantic rule augmentations based on the root rule.

Each augmentation must be defined by:
1. an augmentation intent
2. a logical operator
3. a modification position

Augmentation intents:
{Specific Semantic Augmentation Description (See Below)}

Logical operators:
- AND: add an extra mandatory item
- OR: add an alternative item
- AND_NOT: add an extra mandatory item in negative format
- OR_NOT: add an alternative item in negative format
- NONE: don't add item

Positions:
- IF-X: add the condition with logical operator X
- THEN-X: add the outcome with logical operator X
- IF-X-THEN-X: add the condition with logical operator X and add the outcome with logical operator X

Instructions:
1. Keep every augmentation concise, natural, and self-contained.
2. Preserve the original content as much as possible unless the target augmentation requires a meaningful shift.
3. Stay within the same domain and action space as the root rule.
4. Do not add absurd, irrelevant, or overly specific content.
5. Make each augmentation clearly usable for downstream scenario-based QA generation.

Generate the following augmentations:
{operators}

Output format:
{
"root_rule": "...",
"in_chinese": "...",
"augmentations": {
"<label>": {
"rule": "...",
"change": "...",
"intent": "...",
"operator": "...",
"position": "...",
"in_chinese": "..."
}
}
}

Only output the JSON object.

Semantic Augmentation Description:

- ABSTRACT: make the added item at a more abstract or generalized level or probabilistic while preserving its core logic. Note that you should not to make an item other than the newly added abstract
- NEGATE: reverse the original condition, outcome, or both
- ADDITIVE: add a supplementary condition, duty, limitation, or consequence. The newly added item must be a refinement of the previous item, which is a case of being included by the previous item
- EXCEPTION: introduce a special case where the original rule does not apply or applies differently, you can use word like 'except', 'unless'
- CONFLICT: Create a new condition and the result/action after satisfying that condition. The if-end and then-end of the new rule do not need to include items from the original rule. Give this new rule the highest priority, and once it is satisfied, the original rule will not be executed. Note that this high-priority rule must be integrated into the original rule without changing the meaning of the original rule, and the new rule must still satisfy both the high-priority and conflict requirements. The output must, must, must include both the original rule and the new rule simultaneously. You can use words like 'but', 'otherwise'.
- IRONCLAD: The newly added items after "then" must be adhered to. Note that only the newly added items need to reflect this requirement of adherence; do not change the original ones.

Figure 21: Rule augmentation prompt.

Dependency Plan Construction Prompt

Create the dependency topology for an multi-stage scenario QA instance.

This prompt controls only Stage 1: topology planning.

Strict task boundary:
- Inspect the provided relevant rules only to decide a plausible reasoning graph.
- Do not generate or describe trigger facts, blocker facts, scenario fragments, hidden sub-questions, intermediate conclusions, final conclusions, dependency reasons, semantic summaries, or examples of transferred content.
- Do not explain why a dependency works. Stage 2 will create all semantic and logical content for each step.

Topology goals:
- Use every provided relevant rule exactly once as a rule-bearing step.
- Preserve original rule IDs and cite them only as R(<rule_id>).
- Choose the order and graph from rule relationships, not input order.
- Build a connected DAG with varied reasoning structure. Avoid a single straight chain and avoid a pure star/fan-in graph where all rule steps stay independent until the final step.
- Prefer one or two roots, at least two dependent non-final rule steps, at least one branch, and at least one non-final convergence point.
- You may add one rule_id null final aggregation step when it helps combine earlier branches.
- Every dependency field must contain step_id values only, never rule IDs.
- The final step (the step that produces the overall answer) must have at least two incoming dependencies. That is, its depends_on_step_ids must contain at least two step IDs, and no other step may feed into it.
- Relaxed complexity: simpler but valid topologies (e.g., two parallel branches converging on a final step) are acceptable as long as they avoid a single chain and use all provided rules. Do not force deep nesting or complex branching if it makes planning unreliable.

Forbidden Stage 1 content:
- No trigger facts.
- No intermediate conclusions.
- No dependency rationales or dependency reasons.
- No topology_notes, dependency_summary, rule summaries, or final-answer descriptions.
- No paraphrase of rule text or then-side/consequent outcomes.
- The output must not contain any semantic content fields such as 'dependency_rationales', 'rationales', 'summaries', or 'conclusions'. Only structural fields are allowed.

Output requirements:
- Return only JSON.
- The only graph source is dependency_steps.
- Each step must include exactly these fields: step_id, rule_id, reasoning_role, branch_label, depends_on_step_ids, feeds_into_step_ids.
- The rule_id field is required for every step. For aggregation steps that do not correspond to a single rule, set rule_id to null. Missing rule_id is invalid.
- Each depends_on_step_ids and feeds_into_step_ids item must be a step_id.
- If final aggregation exists, set rule_id to null.

Return schema:
```
{
 "ordered_rule_ids": [],
 "reasoning_topology": "branched_convergence | hub_and_spoke | independent_checks_then_interaction | exception_gate | mixed",
  "dependency_steps": [
   {
    "step_id": "",
    "rule_id": "",
    "reasoning_role": "independent_root | branch_source | branch_check | convergence | exception_gate | final_aggregation",
    "branch_label": "",
    "depends_on_step_ids": [],
    "feeds_into_step_ids": []
   },
   {
    "step_id": "",
    "rule_id": "",
    "reasoning_role": "",
    "branch_label": "independent_root | branch_source | branch_check | convergence | exception_gate | final_aggregation",
    "depends_on_step_ids": [],
    "feeds_into_step_ids": []
   }
  ]
}
```

Question seed:
{question_seed}

Relevant rules:
{rule_text},

Return only one valid JSON object.

Figure 22: Dependency plan construction prompt.

Sub-Scenario Generation Prompt

Create one enriched dependency step for an multi-stage scenario QA instance.

This prompt controls only Stage 2: semantic and logical completion for one topology step.

Task boundary:
- Work only on the current dependency step.
- Do not redesign step_id, rule_id, depends_on_step_ids, or feeds_into_step_ids.
- Stage 1 supplied topology only. You must now create the concrete story facts, hidden sub-question, intermediate conclusion, downstream contribution, and dependency rationales.
- Use prerequisite step context when the current step has depends_on_step_ids.
- Use downstream step rule context to make the current step's conclusion genuinely useful later.

Core goals:
- Write a natural scenario_fragment of about 200 words for substantive rule-bearing steps. It should feel like part of a larger continuous story, with documents, dates, messages, decisions, timing pressure, and harmless distractions.
- If the current step has a target rule, embed all needed facts naturally in the fragment without copying or closely paraphrasing rule wording.
- If rule_id is null, treat the step as aggregation: synthesize prior intermediate conclusions without inventing a new rule.
- Put rule-governed conclusions only in hidden fields such as intermediate_conclusion and downstream_contribution, never in public-facing story prose.
- Make shallow keyword matching tempting but insufficient.

Dependency rationales:
- Every output step must include dependency_rationales.
- If depends_on_step_ids is empty, dependency_rationales must be [].
- For each id in depends_on_step_ids, include exactly one dependency_rationales item.
- Each rationale must use depends_on_step_id as a step id, never a rule id.
- Each rationale must explain an irreducible dependency: without the prior step's intermediate_conclusion, the current step's intermediate_conclusion cannot be derived correctly.
- A dependency is invalid if it is merely helpful, chronological, thematic, based on shared background facts, or still derivable from the current step's own facts.

Leakage prevention:
- scenario_fragment must contain only concrete story facts.
- Do not include rule IDs, rule labels, hidden sub-questions, if/then clauses, copied rule text, close paraphrases, or direct statements of then-side/consequent outcomes.
- Avoid abstract rule terms such as exception, condition satisfied, prohibited, duty, permission, eligibility, validity, priority, and final outcome unless they appear as ordinary story facts rather than rule conclusions.
- After drafting, rewrite any phrase that resembles the rule text or exposes the rule outcome.

Output requirements:
- Return only JSON.
- Use canonical R(<rule_id>) only in hidden fields when a rule_id exists.
- Keep generated fields useful enough for Stage 3 to weave into one natural story.
- Do not contradict prerequisite steps.

Return schema:
```
{
 "step_id": "",
 "rule_id": "",
 "scenario_fragment": "About 200 words of detailed story for this step.",
 "hidden_sub_question": "Internal sub-question for generation only.",
 "trigger_facts": ["explicit facts embedded in the fragment"],
 "exception_or_blocker_facts": ["explicit exception or blocker facts, or none"],
 "intermediate_conclusion": "hidden conclusion from this step's rule, or aggregation conclusion when rule_id is null",
 "dependency_rationales": [
   {
    "depends_on_step_id": "",
    "prior_conclusion_used": "the hidden conclusion from the prerequisite step",
    "why_current_step_depends_on_it": "specific reason the current conclusion cannot be derived without that prior conclusion",
    "if_dependency_removed": "what becomes impossible, underdetermined, or incorrect if that prior conclusion is removed"
   }
 ],
 "downstream_contribution": "how this step's conclusion should influence downstream step ids"
}
```

Question seed:
{question_seed}

Current dependency step:
{current_dependency_step}

Target rule for the current step:
{target_rule_text}

Already generated dependency context for prerequisite steps:
{dependency_context},

Downstream step rule context that this step should support:
{downstream_steps}

Return only one valid JSON object.

Figure 23: Sub-scenario generation prompt.

Final Synthesis Prompt (Part-1)

Synthesize enriched dependency steps into one scenario QA instance.

This prompt controls only Stage 3: final public scenario, one final question, reference answer, and QA-specific answer judge rubric.

Input boundary:
- Use only the enriched dependency_steps, the question_seed, and the answer-quality rubric reference provided by the wrapper.
- The enriched dependency_steps already contain topology, rule text for review, scenario fragments, hidden sub-questions, trigger facts, intermediate conclusions, dependency rationales, and downstream contributions.
- Do not ask for a separate rule list or a separate sub-scenarios list.
- Do not regenerate answer_key.rule_logic_chain. Leave it absent or empty; the pipeline will fill it from the enriched dependency steps.

Final scenario goals:
- Weave all scenario fragments into one coherent, self-contained narrative.
- Produce a single continuous narrative with natural paragraph breaks only. Do not use any mechanical segmentation: no sub-scenario headings, no temporal markers like "Meanwhile" or "On the morning of...", no step/part/scene labels, and no visible newline markers.
- Interleave facts naturally so the reader cannot infer how many construction steps existed.
- Include all facts needed for one unique answer, plus harmless distractions, without contradiction.
- Make cross-rule dependencies easier to trace by ensuring the narrative naturally embeds all relevant rule triggers without explicitly enumerating rules. Clarify key facts and reduce irrelevant distractions to lower difficulty while maintaining discrimination.

Public leakage constraints:
- final_scenario and final_question must not contain rule IDs, rule labels, if/then clauses, copied rule text, close paraphrases, hidden sub-questions, or rule-governed conclusions.
- Do not directly state any then-side/consequent outcome, including duties, permissions, prohibitions, eligibility statuses, validity outcomes, priority results, exception results, or final rule outcomes.
- Conclusions belong only in answer_key, rule_applications, and answer_judge_rubric.

Final question:
- Ask exactly one integrated question that requires synthesis across the entire narrative.
- Do not use bullets, numbering, multiple questions, or exposed sub-questions. The question must not ask for two distinct issues (e.g., compliance and resolution) but instead a single holistic determination.

Answer key:
- Include decomposition, rule_applications, and final_answer.
- Cite every relevant rule in canonical R(<rule_id>) form.
- Do not include rule_logic_chain, or set it to [] only. The pipeline will replace it with the enriched Stage 2 chain.

QA-specific answer_judge_rubric:
- Generate answer_judge_rubric for judging tested model answers on this exact QA.
- Use the provided answer-quality rubric reference as mandatory coverage: every reference dimension must be represented. Do not include any dimensions not in the reference, especially construction-stage dimensions, system-computed rule recall, or rule-application precision.
- You may refine each dimension's binary_criteria so it is specific to this QA's facts, rules, dependencies, and final answer.
- total_points must be 100.
- Every dimension must be all-or-nothing: score 0 unless all criteria in that dimension are met. Each dimension must have a single binary_criterion (not a list of criteria).
- Every dimension must have positive points.

Output requirements:
- Return only JSON.
- Use R(<rule_id>) only in hidden/reference fields.

Return schema:
```
{
 "sample_type": "scenario_rule_qa",
 "question_seed": "seed_name",
 "final_scenario": "One coherent scenario.",
 "final_question": "One single integrated question.",
 "answer_key": {
   "decomposition": ["high-level issue decomposition"],
   "rule_logic_chain": [],
   "rule_applications": [
     {"rule_id": "", "fact": "scenario fact", "conclusion": "rule-governed result"}
   ],
   "final_answer": "answer citing every R(...) rule id."
 },
 "answer_judge_rubric": {
   "rubric_type": "answer_quality",
   "total_points": 100,
   "scoring_policy": "Each dimension is all-or-nothing. Award either 0 or the dimension's full points, and always provide a brief factual basis.",
   "dimensions": [
     {
       "key": "dimension_key",
       "name": "Dimension name",
       "points": 10,
       "binary_criteria": "QA-specific all-or-nothing criterion",
       "pass_condition": "Award either 10 or 0; no partial credit."
     }
   ]
 }
}
```

Question seed:
{question_seed},

Answer-quality rubric reference. Generate a QA-specific answer_judge_rubric that preserves every dimension but may refine binary_criteria for this QA:

Figure 24: Final synthesis prompt (part 1).

Final Synthesis Prompt (Part-2)

```
{
 "rubric_type": "answer_quality",
 "total_points": 100,
 "scoring_policy": "Each dimension is all-or-nothing. Award either 0 or the dimension's full points, and always provide a brief factual basis. Rule recall and rule-application precision are measured separately and must not be included as rubric dimension scores.",
 "dimensions": [
   {
    "key": "question_understanding",
    "name": "Question understanding",
    "points": 10,
    "criteria": [
      "The answer identifies the exact outcome requested by the final question.",
      "The answer does not replace the requested determination with a nearby but different task.",
      "The answer keeps the scope of the question clear throughout the reasoning."
    ]
   },
   {
    "key": "issue_decomposition",
    "name": "Issue decomposition",
    "points": 10,
    "criteria": [
      "The answer decomposes the final question into all necessary sub-issues.",
      "The decomposition connects each sub-issue to concrete scenario facts.",
      "The answer does not introduce irrelevant sub-issues that drive the final answer."
    ]
   },
   {
    "key": "citation_format_compliance",
    "name": "Citation format compliance",
    "points": 8,
    "criteria": [
      "Every rule reference used in reasoning appears in canonical R(<rule_id>) form.",
      "The answer does not rely on bare numeric IDs, uncited rule references, or invented rule IDs.",
      "Rule citations appear at the point where the rule is applied, not only in a final list."
    ]
   },
   {
    "key": "rule_grounded_reasoning",
    "name": "Rule-grounded reasoning",
    "points": 17,
    "criteria": [
      "The answer explains how cited rules interact with concrete scenario facts.",
      "The answer states the rule-governed conclusion drawn at each substantive rule application.",
      "The answer's reasoning is explicit enough to audit without relying on hidden assumptions."
    ]
   },
   {
    "key": "dependency_chain_alignment",
    "name": "Dependency chain alignment",
    "points": 18,
    "criteria": [
      "The answer follows the essential dependencies in the reference answer_key.rule_logic_chain.",
      "The answer passes required intermediate conclusions from prior steps into later reasoning where the reference chain requires it.",
      "The answer does not treat genuinely dependent rule checks as independent checks.",
      "The answer does not add extra logical dependencies that change the final conclusion."
    ]
   },
   {
    "key": "exception_conflict_handling",
    "name": "Exception, conflict, and blocker handling",
    "points": 12,
    "criteria": [
      "The answer correctly handles exceptions, blockers, overrides, priority rules, and conflicts required by the reference answer.",
      "The answer correctly identifies which prior conclusions unlock, block, or modify later rule checks.",
      "Any exception, blocker, priority, or conflict condition central to the final answer is resolved correctly."
    ]
   },
   {
    "key": "intermediate_conclusion_quality",
    "name": "Intermediate conclusion quality",
    "points": 10,
    "criteria": [
      "Intermediate conclusions match the reference rule-governed conclusions without overgeneralizing.",
      "Each intermediate conclusion is tied to the scenario fact and cited rule that supports it.",
      "The answer does not invent extra duties, permissions, prohibitions, eligibility statuses, validity results, or outcomes."
    ]
   },
   {
    "key": "final_answer_consistency",
    "name": "Final answer consistency",
    "points": 12,
    "criteria": [
      "The final answer follows from the answer's stated facts, cited rules, intermediate conclusions, and aggregation reasoning.",
      "The final answer directly answers the public final question.",
      "The final answer includes all required applicable outcomes and omits unsupported outcomes."
    ]
   },
   {
    "key": "no_external_facts",
    "name": "No external facts",
    "points": 3,
    "criteria": [
      "The answer does not rely on facts, policies, legal rules, world knowledge, or assumptions outside the scenario and provided rules."
    ]
   }
 ]
}
```

Enriched dependency steps:
{dependency_steps}

Return only one valid JSON object.

Figure 25: Final synthesis prompt (part 2).

QA Quality Judge Prompt (part-1)

```
Act as a strict dataset quality judge for the scenario QA sample below. Every dimension score must be either 0 or that dimension's full points, with a brief factual basis.

QA-quality scoring rubric:
{
 "total_points": 100,
 "scoring_policy": "Each dimension is all-or-nothing. Award either 0 or the dimension's full points, and always provide a brief factual basis. This rubric evaluates the generated QA item after the dependency plan has already passed its gate and been frozen; it does not evaluate dependency-plan topology or any tested model answer.",
 "dimensions": [
   {
    "key": "sub_scenario_fact_and_story_support",
    "name": "Sub-scenario fact and story support",
    "stage": "sub_scenario",
    "points": 18,
    "criteria": [
     "Each enriched step contains a natural scenario_fragment with explicit facts needed for that step's rule or aggregation.",
     "Substantive rule-bearing fragments are about 200 words and include plausible story detail and harmless distractions.",
     "Fragments remain coherent with prerequisite steps and useful for final synthesis.",
     "Every step contains trigger_facts, intermediate_conclusion, downstream_contribution, and hidden_sub_question where applicable."
    ]
   },
   {
    "key": "sub_dependency_irreducibility",
    "name": "Sub-scenario dependency irreducibility",
    "stage": "sub_scenario",
    "points": 20,
    "criteria": [
     "Every step includes a dependency_rationales array, even if empty.",
     "For every depends_on_step_ids item, there is exactly one dependency_rationales item using that prior step id.",
     "Each dependency rationale identifies the prior conclusion used, explains why the current conclusion cannot be derived without it, and states what fails if it is removed.",
     "Any dependency that is merely helpful, chronological, thematic, based on shared background facts, or derivable from current-step facts causes this dimension to receive 0.",
     "The judge must not ask to change step ids, rule ids, depends_on_step_ids, feeds_into_step_ids, or dependency topology; those fields are frozen after the dependency gate."
    ]
   },
   {
    "key": "sub_scenario_non_leakage",
    "name": "Sub-scenario non-leakage",
    "stage": "sub_scenario",
    "points": 12,
    "criteria": [
     "Scenario fragments do not expose rule citations, labels, hidden sub-questions, copied rule text, or close paraphrases.",
     "Scenario fragments do not directly state rule then-side/consequent outcomes.",
     "Review-only fields such as rule and scenario_fragment inside answer_key.rule_logic_chain are not schema violations by themselves; judge only whether their public-facing content leaks."
    ]
   },
   {
    "key": "final_scenario_natural_synthesis",
    "name": "Final scenario natural synthesis",
    "stage": "final_synthesis",
    "points": 14,
    "criteria": [
     "The final scenario reads as one coherent natural narrative with smooth transitions.",
     "It is not mechanically segmented by step, sub-scenario, rule, entity list, headings, or visible /n or \\n markers.",
     "A small number of natural paragraphs is acceptable only when it does not reveal construction stages.",
     "The scenario is self-contained, fair, and uniquely answerable."
    ]
   },
   {
    "key": "final_single_integrated_question",
    "name": "Single integrated final question",
    "stage": "final_synthesis",
    "points": 8,
    "criteria": [
     "The final question is exactly one integrated public question.",
     "It is not a numbered list, bullet list, multi-part prompt, or sequence of exposed sub-questions.",
     "It asks only the relevant final question and does not reveal hidden sub-questions."
    ]
   },
   {
    "key": "final_answer_key_completeness",
    "name": "Final answer key completeness",
    "stage": "final_synthesis",
    "points": 12,
    "criteria": [
     "The answer key includes decomposition, rule_applications, and final_answer.",
     "The answer key includes the pipeline-filled answer_key.rule_logic_chain derived from enriched Stage 2 steps.",
     "The final_answer and rule_applications cite all relevant rules in canonical R(<rule_id>) form.",
     "The answer key is sufficient for a judge to verify the unique final answer."
    ]
   },
```

Figure 26: Scenario question answering quality judgment prompt (part 1).

QA Quality Judge Prompt (part-2)

```
   {
    "key": "generated_answer_rubric_completeness",
    "name": "QA-specific answer rubric completeness",
    "stage": "final_synthesis",
    "points": 8,
    "criteria": [
     "The QA includes answer_judge_rubric rather than relying only on an external answer rubric.",
     "The generated answer_judge_rubric totals 100 points, has only positive point dimensions, and uses all-or-nothing criteria.",
     "The generated answer_judge_rubric covers every external answer-quality dimension while adapting criteria to this QA.",
     "The generated answer_judge_rubric does not add QA-construction dimensions, rule recall, or rule-application precision as answer-quality dimensions."
    ]
   },
   {
    "key": "public_non_leakage",
    "name": "Public scenario/question non-leakage",
    "stage": "final_synthesis",
    "points": 8,
    "criteria": [
     "The public final_scenario and final_question do not reveal rule IDs, rule labels, copied or closely paraphrased rule text, visible if/then clauses, or hidden sub-questions.",
     "They do not directly state any rule-governed conclusion, including duties, permissions, prohibitions, eligibility, validity, priority, exception results, or final outcomes."
    ]
   }
  ]
}

Rule-text leakage check:
- final_scenario and final_question are public text shown to the evaluated model.
- They must not include R(<rule_id>) citations, rule labels, copied rule sentences, close paraphrases of rule clauses, or visible rule-like if/then instructions.
- They must also not directly state the then-side/consequent of any rule, including hidden duties, permissions, prohibitions, eligibility statuses, validity outcomes, priority results, exception results, or final rule outcomes.
- It is acceptable for the public text to contain natural story facts that trigger a rule, but not to expose the rule itself.
- If leakage exists, include a weakness and set the related dimension score to 0.

All-or-nothing scoring reminder:
- Every dimension score must be either 0 or that dimension's full point value.
- Every dimension_scores entry must include score, max_points, and reason.
- The reason must briefly cite the concrete generated-QA evidence supporting the score.

Return schema:
{
  "dimension_scores": [],
  "total_score": "",
  "strengths": [],
  "weaknesses": [],
  "fixes": [],
}

Rules:
{rules_text}

Scenario QA instance:
{qa_json}
```

Figure 27: Scenario question answering quality judgment prompt (part 2).

QA Template

```
Answer the question based only on the given scenario and the provided rules. Write a step-by-step natural-language reasoning answer following these requirements:

## Requirements:
1. Do not assume any facts outside the scenario.
2. Treat the rules below as the complete and only rule set for this question. Use only relevant listed rules; do not invent rule IDs or use unlisted rules.
3. Begin by analyzing the question in prose: identify the exact requested outcome and decompose it into necessary sub-issues.
4. Reason step by step in natural language. Whenever you apply a rule, cite it in canonical form R(<rule_id>), state the scenario fact that triggers it, explain the inferred result, and describe how it affects later reasoning.
5. If a step depends on an earlier conclusion, state that dependency explicitly.
6. Identify and use all rules needed to answer the question, and cite only rules that are actually applied in the reasoning.
7. End with a clear final answer that directly answers the question.

## Rules:
{rules}

## Scenario:
{final_scenario}

## Question:
{question}
```

Figure 28: Question answering template.

Answer Judge Template

You are a strict answer quality judge.

Task:
Evaluate the tested model's answer against the reference answer context, scenario, relevant rules, and answer quality rubric.

Inputs:
Answer quality scoring rubric:
{answer_judge_rubric_json}

Reference answer context:
{reference_context_json}

Tested model answer:
{model_answer}

Output JSON schema:
{output_schema_json}

Strict judging principles:
1. Judge strictly according to the provided answer quality rubric.
2. Each dimension in dimension_scores must correspond exactly to one rubric dimension.
3. Do not add, merge, split, reinterpret, or rename rubric dimensions.
4. Do not award points based on partially related content, implied reasoning, or charitable interpretation.
5. If the tested answer does not clearly and explicitly satisfy a rubric dimension, assign 0 for that dimension.
6. In ambiguous cases, assign 0.
7. Each rubric dimension is all or nothing. The score for each dimension must be either 0 or its full max_points value.
8. The top level total_score must be the sum of all scores in dimension_scores.

Rule citation and application:
1. Canonical citations in the form R(<rule_id>) count as valid rule citations.
2. Bare numeric IDs do not count as valid rule citations.
3. For precision related fields, judge whether each cited R(<rule_id>) is applied to the correct scenario fact and produces the correct result.
4. Penalize unsupported external assumptions, missing intermediate conclusions, broken reasoning dependencies, incorrect rule applications, and final answers that do not follow from the stated reasoning.

Required output:
1. Output only valid JSON that conforms to the provided output schema.
2. Every dimension_scores entry must include:
   a. score
   b. max_points
   c. a concise factual reason grounded in the tested answer and the reference answer context
3. Do not output a top level field named score.

Figure 29: Answer judge template.