

PaperGraph-Bench: Graph-Guided Benchmark Construction and Multi-Turn Evaluation for Scientific Paper Reading

Rao Fu, Qing Dai, Kaitai Ou-yang, Kang He, Fei Li, Donghong Ji, Chong Teng*

Key Laboratory of Aerospace Information Security and Trusted Computing,
Ministry of Education, School of Cyber Science and Engineering, Wuhan University
{furao2003, tengchong}@whu.edu.cn

Abstract

Scientific-paper reading requires models to recover a paper’s core argument, connect claims to textual and multimodal evidence, and avoid unsupported inferences across dialogue turns. We introduce PAPERGRAPH-BENCH, a graph-guided benchmark that turns each paper into a paper-specific evaluation state graph with macro stages, evidence-grounded knowledge components (KCs), verified reasoning edges, multimodal assets, rubrics, and trajectory state. The graph drives macro coverage questions, challenge questions, reasoning threads, and bounded repair turns, while the evaluator tracks KC coverage, challenge failures, hallucination events, and repair outcomes. Experiments on a human-audited prototype with ten models show that coverage, robustness, hallucination type, and repair behavior reveal distinct paper-reading abilities. Our code and current dataset are available at <https://github.com/furao0011/PaperGraphBench>.

1 Introduction

Large language models are increasingly used to read, summarize, and check scientific papers. This requires more than answering isolated questions: a model must recover a paper’s argument, connect claims to textual and multimodal evidence, and respect stated limitations. A model may therefore appear competent on local QA while still missing the main contribution, accepting a false premise about an experiment, or overclaiming beyond the paper’s evidence.

Existing benchmarks cover parts of this problem but rarely evaluate paper understanding as a stateful, multimodal process. Paper QA and claim-verification datasets such as QASPER and SciFact ground questions or claims in research papers (Dasigi et al., 2021; Wadden et al., 2020). Multimodal document and scientific-figure benchmarks

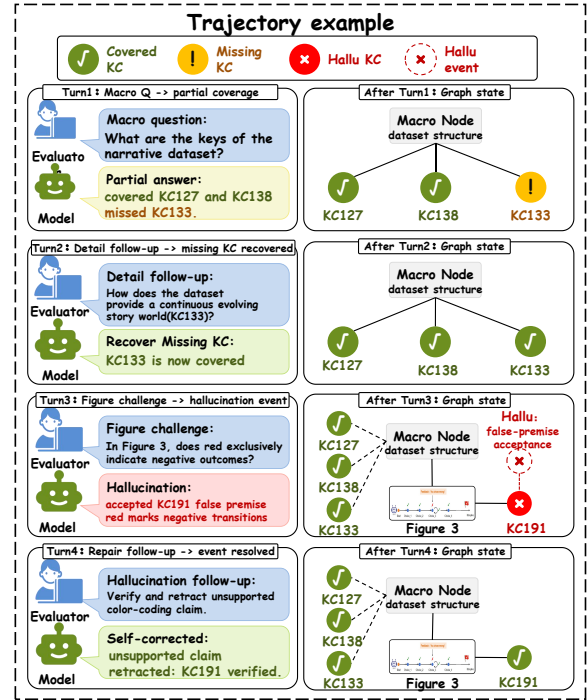


Figure 1: A trajectory example in PAPERGRAPH-BENCH. A missing KC triggers a detail follow-up, while accepting a false figure premise registers a hallucination event and triggers a repair follow-up; each row shows the judged turn and the resulting local graph state.

show that visual evidence is essential (Masry et al., 2022; Ma et al., 2024; Pramanick et al., 2024), while hallucination and LLM-as-judge work makes open-ended evaluation more scalable (Li et al., 2023; Zheng et al., 2023). However, most evaluations still treat a paper as static evidence, leaving unclear whether a model covers the paper’s core knowledge, resists graph-grounded false premises, or repairs errors across dialogue turns.

We identify three forms of evaluation invisibility. First, paper QA often lacks *coverage visibility*: without an explicit paper structure, it is hard to know whether evaluation reaches central claims and reasoning dependencies. Second, multimodal

*Corresponding author.

Work	Paper ground	Graph struct.	Multi modal	Multi turn	Event halluc.
QASPER	✓	–	–	–	–
SciFact	✓	–	–	–	✓
SPIQA	✓	–	✓	–	–
MMLong-Doc	–	–	✓	–	–
HaluEval/MT	–	–	–	✓	✓
Ours	✓	✓	✓	✓	✓

Table 1: Comparison with representative benchmarks. Paper ground denotes scientific-paper-grounded evaluation; Graph struct. denotes explicit paper or knowledge-component graph construction; Event halluc. denotes event-level hallucination tracking.

evidence is often treated as a separate modality rather than part of the paper’s argument, limiting *evidence-bound robustness* against prompts that conflict with figures, tables, or results. Third, hallucination is often a coarse answer-level label, while paper dialogue needs *trajectory visibility*: errors should stay linked to the paper structure and to later turns where they may be repaired, amplified, or reintroduced.

We propose PAPERGRAPH-BENCH, a graph-guided framework for multimodal, multi-turn scientific-paper evaluation. For each paper, we build a paper graph with macro stages, textual and multimodal KCs, verified reasoning edges, and graph-derived reasoning threads. This graph drives macro questions for coverage, challenge questions for false-premise, overclaim, and wrong-relation failures, and thread questions for multi-step reasoning. During evaluation, a scheduler asks graph-linked questions, a judge assesses coverage and hallucination events with rubrics, and the state updater records which KCs are covered, which unsupported claims are accepted, and whether later turns repair or amplify earlier mistakes. Our main contributions are summarized as follows:

- We propose a graph-guided pipeline for turning scientific papers into benchmark instances, including macro structure, textual and multimodal KCs, verified reasoning edges, and graph-grounded question sets.
- We introduce a trajectory-driven multi-turn evaluation framework that measures knowledge coverage, challenge failure, hallucination events, modality-specific robustness, and dialogue efficiency.
- We construct a human-filtered benchmark and

define metrics that expose failure modes hidden by isolated single-turn QA.

2 Related Work

2.1 Paper Understanding Benchmark

Scientific-paper understanding has been evaluated through paper-grounded QA and evidence inference (Dasigi et al., 2021; Lehman et al., 2019), scientific claim verification (Wadden et al., 2020; Wang et al., 2025a), retrieval-augmented paper QA (Lála et al., 2023), and recent multimodal or multi-document scientific QA (Li et al., 2024). Multimodal document benchmarks cover scanned documents and infographics (Mathew et al., 2021, 2022), plots, charts, and scientific tables (Methani et al., 2020; Masry et al., 2022; Zhang et al., 2025; Wang et al., 2024), long documents and scientific figures (Ma et al., 2024; Hsu et al., 2021; Pramanick et al., 2024), and cross-source reasoning in papers (Tian et al., 2025). These works motivate paper-grounded and multimodal evaluation; PAPERGRAPH-BENCH instead makes the evaluation unit a paper-specific graph that links KCs, multimodal assets, reasoning edges, rubrics, and state updates.

2.2 Grounded Hallucination Benchmark

Hallucination and factuality benchmarks include truthfulness tests (Lin et al., 2022), response-level hallucination evaluation (Li et al., 2023; Chen et al., 2023), retrieval-grounded factuality (Niu et al., 2024), long-form claim decomposition (Min et al., 2023; Wei et al., 2024), hard hallucination cases (Fan et al., 2026), and KG-grounded evaluation (Sansford et al., 2024; Robertson et al., 2026). We follow their move toward grounded and decomposed labels, but attach each hallucination event to a paper graph node, edge, or forbidden claim, enabling paper-specific labels such as contradicted KC, false-premise acceptance, wrong-relation acceptance, and review regression.

2.3 Dynamic Evaluation Benchmark

Dynamic evaluation work includes LLM-as-judge and arena protocols for scalable open-ended evaluation (Zheng et al., 2023; Chiang et al., 2024), dialogue hallucination and fine-grained multi-turn dialogue benchmarks (Chen et al., 2024; Bai et al., 2024), structured turn-dependency tests (Li et al., 2025), long-context and long-term-memory benchmarks (Bai et al., 2025; Wan and Ma, 2025), and

Statistic	Total	Avg./paper
Papers	12	—
Paper graphs	12	—
KC nodes	1727	143.9
Multimodal KC nodes	459	38.2
Reasoning edges	1804	150.3
Macro questions	97	8.1
Challenge questions	238	19.8
Text challenges	120	10.0
Multimodal challenges	118	9.8
Thread question seeds	183	15.2

Table 2: Dataset-scale statistics for the 12-paper benchmark.

adaptive graph-based test generation (Zhang et al., 2024). Our graph is neither a general knowledge graph nor a retrieval-evidence graph; it is a paper-specific evaluation state graph whose nodes, edges, rubrics, and state updater jointly support coverage tracking, challenge generation, hallucination localization, and repair measurement over turns.

3 Our Benchmark

3.1 Benchmark Overview

PAPERGRAPH-BENCH is a compact but deep benchmark for scientific-paper understanding. We focus on NLP papers because they combine textual claims, experimental evidence, multimodal assets, and method/result dependencies. Each benchmark instance corresponds to one paper and contains a paper graph, graph-linked questions, multimodal asset references, KC rubrics, forbidden claims, and trajectory-state metadata:

$$B_P = (G_P, Q_{\text{macro}}, Q_{\text{chal}}, Q_{\text{thread}}, R_P), \quad (1)$$

where R_P denotes rubrics and evaluation metadata. Macro questions test core coverage, challenge questions test robustness to misleading prompts, and graph-derived thread seeds test multi-step reasoning.

Figure 3 outlines the construction flow from raw paper to graph-linked evaluation package. The following subsections detail graph construction, question generation, and the multi-turn evaluation protocol.

Table 2 reports aggregate statistics, but the benchmark is intentionally compact: we use twelve recent, low-citation papers, most of which postdate the evaluated models’ knowledge cutoffs. This reduces memorized paper-specific knowledge as an extra variable; Appendix A.6 and Appendix A.1 detail the paper selection and model cutoffs.

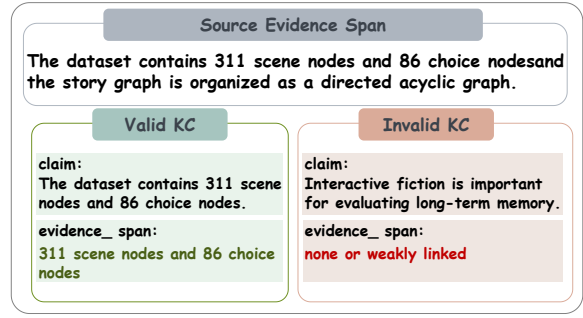


Figure 2: KC candidate examples. A valid KC is atomic, evidence-grounded, and evaluable; an invalid candidate is rejected when it is generic or weakly supported.

3.2 Paper Graph Construction

Given a paper $P = (T, A_{\text{mm}})$, where T is the paper text and A_{mm} contains figures and tables, we construct a paper-specific graph

$$G_P = (M, K, E, A_{\text{mm}}), \quad (2)$$

where M is the macro spine, K is a bank of textual and multimodal knowledge components (KCs), and E is a set of verified reasoning edges. The graph is an evaluation scaffold rather than a unique ontology of the paper: it provides reviewed, evidence-grounded targets and relations for question generation, judging, and state updates.

Macro spine. The macro spine is extracted from the paper structure rather than a fixed Introduction–Method–Experiment taxonomy. Each macro node records its title, role, summary, source sections, and expected reader question; after filtering, the released papers contain roughly eight macro questions covering their major argumentative stages.

Knowledge components. For each macro-aligned textual unit or multimodal asset, we extract KC candidates. A KC is an atomic, evidence-grounded claim that can be verified from a source span or asset and used as a judging target. Candidates are rejected or rewritten when they merge independent facts, state generic background without paper-specific evidence, or require unsupported inference. Figure 2 illustrates the candidate fields and validity decision. Textual and multimodal candidates are then normalized into a shared KC bank:

$$K = \Phi(C_{\text{text}} \cup C_{\text{mm}}), \quad (3)$$

where Φ attaches evidence, rubrics, modality metadata, duplicate links, and review flags. Appendix B gives the construction details, and Appendix F collects concrete graph-object instances.

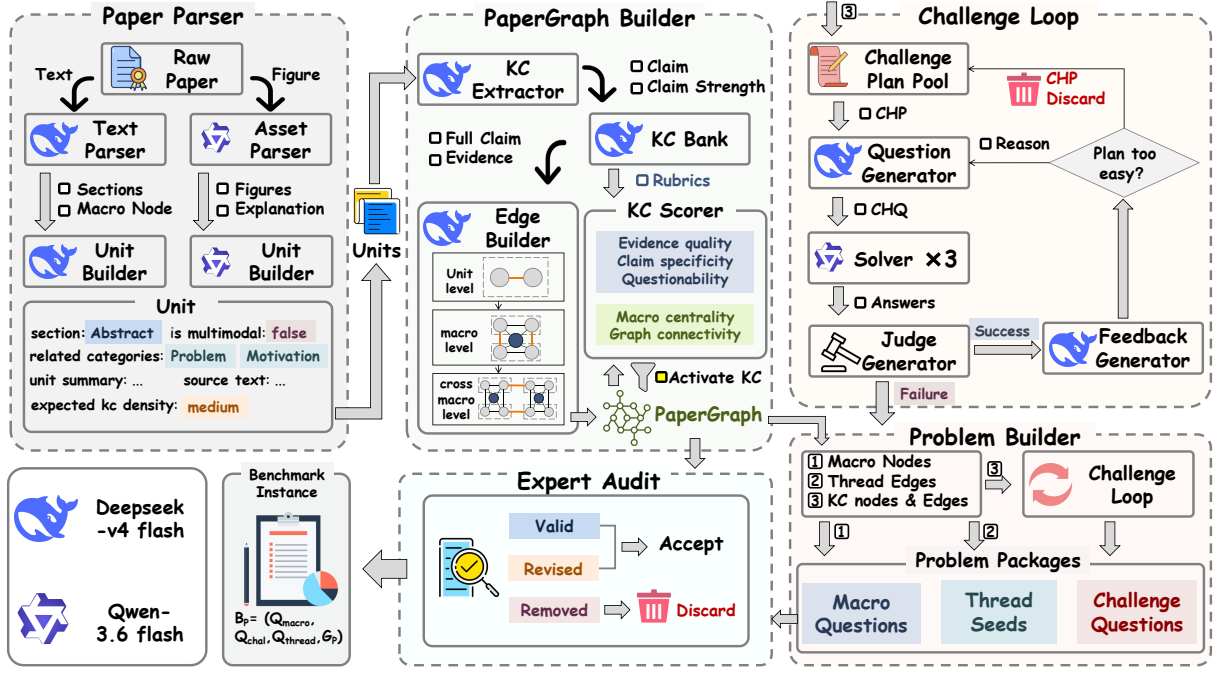


Figure 3: Dataset construction framework. A raw paper is parsed into text and multimodal units, converted into a paper graph with KCs, reasoning edges, rubrics, and asset nodes, and used to generate macro questions, challenge questions, and thread seeds.

Reasoning edges. We construct directed reasoning edges between KCs at four scopes: unit, macro, cross-macro, and thread. These edges capture local dependencies, within-stage relations, nearby cross-stage links such as method–result support, and longer paths used for multi-turn seeds. Downstream, verified edges support wrong-relation challenges, thread generation, and dependency-aware judging.

Each candidate edge contains a source KC, target KC, relation type, evidence packet, and rationale. A verifier checks whether the source, target, relation label, direction, and evidence are jointly supported. Only valid or relabeled edges enter the final graph; invalid, insufficient-context, and wrong-direction candidates are excluded. Appendix B.3 gives the relation inventory and verifier decisions.

Active KC selection. The KC bank is larger than the target set used by main questions, so we select active KCs for core coverage evaluation. Each KC receives a priority score over normalized features $\mathcal{F}$, including evidence quality, specificity, questionability, macro centrality, and graph connectivity:

$$s(k) = \sum_{\ell \in \mathcal{F}} w_\ell f_\ell(k), \quad (4)$$

$$K^* = \text{SelectTop}(K_{\text{text}}, s, \rho),$$

where ρ denotes a simple per-macro coverage rule that prevents targets from concentrating in a few macro nodes. The score $s(k)$ is a prioritization score, not a truth score: unselected KCs remain in the graph for challenges, repairs, and state tracking. In the main configuration, macro coverage is computed over text-default active KCs for comparability, while multimodal KCs remain first-class graph nodes for multimodal challenges and state updates. Appendix B.4 details the weights and Appendix B.5 summarizes the multimodal configuration.

3.3 Question Generation

Macro questions. Each macro node is paired with its active KCs to generate one main question. Storing the target KC set lets the evaluator distinguish complete answers, partial answers, and unsupported additions.

Challenge questions. Challenge questions are produced by a solver-in-the-loop filter over separate text and multimodal plan pools. Each plan specifies a target KC or edge, failure type, preserved evidence, and forbidden premise or relation. A generator writes the question, a validator checks answerability and faithfulness, and solver models attempt it; we retain the challenge only when it is valid and at least one solver exhibits the

intended failure. This filter removes trivial prompts whose false premise is too obvious, while keeping plausible hard cases grounded in a human-audited package. If all solvers answer correctly, the plan is either discarded as too easy or sent back for surface-question revision.

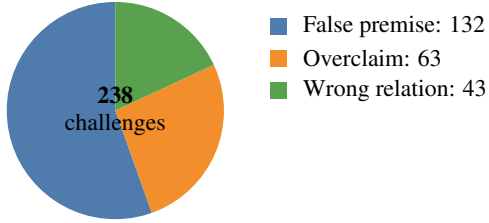


Figure 4: Challenge type distribution after solver-based filtering.

We do not force uniform challenge-type balance. As shown in Figure 4, the final set reflects accepted hard cases under the filtering criterion, so aggregate challenge failure is interpreted as a mixture over false-premise, overclaim, and wrong-relation failures.

Thread seeds. Thread reasoning edges are converted into scheduler-ready seeds with premise, evidence, bridge, or review roles. Rather than forming a static question list, they are consumed according to dialogue progress and graph state to support follow-ups, related challenges, and final review.

3.4 Evaluation Framework

Given $B_P = (G_P, Q_{\text{macro}}, Q_{\text{chal}}, Q_{\text{thread}}, R_P)$, evaluation runs as a graph-guided dialogue. The target model receives the paper context, dialogue history, and optional figure or table inputs; after each answer, a judge compares the response with KC rubrics and forbidden claims, and the evaluator updates graph state.

Stateful protocol. At turn t , the scheduler selects a graph-linked question q_t from state S_{t-1} . The model answers, the judge returns a structured result, and the updater records coverage, hallucination, thread, and challenge state:

$$\begin{aligned} a_t &= f_\theta(q_t, P, h_{<t}, A_t), \\ j_t &= J(a_t, q_t, G_P, R_P, h_{<t}), \\ S_t &= U(S_{t-1}, j_t). \end{aligned} \quad (5)$$

Here $h_{<t}$ is dialogue history and A_t denotes optional multimodal assets. The state S_t tracks KC

status, macro progress, coverage gaps, hallucination events, challenge states, thread states, and repair tasks.

Dialogue scheduling. Evaluation proceeds macro by macro. Each stage starts with a macro question; missing target KCs trigger detail follow-ups, while hallucinations or accepted misleading premises trigger hallucination follow-ups. Repair tasks are bounded to three turns. The scheduler then samples ready thread seeds, inserts macro- or thread-grounded challenges, and ends with review questions over previously covered content. Appendix Table 18 summarizes the turn types.

Judging and state updates. The judge outputs covered KCs, missing KCs, hallucination events, challenge and thread results, confidence, and a recommended next action. Coverage and hallucination are separated: a response may cover target KCs while adding unsupported content, or miss KCs without hallucinating. The updater lights covered KCs, records gaps, registers hallucination events, and links follow-ups to the triggering gap or event.

All turns are written to `dialogue_trajectory.json`, which stores the question type, target KCs, answer, judge result, multimodal assets, repair context, and realized next action. Appendix C.1 summarizes package construction, Appendix D.2 describes trajectory records, and Appendix F collects concrete examples.

4 Quality Control

Because PAPERGRAPH-BENCH uses LLM-assisted construction and judging, quality control focuses on decisions that directly affect reported scores. Two expert annotators independently reviewed the audited items, and disagreements were adjudicated before release.

4.1 Construction Validity

We conduct full double-expert audits over candidate graph objects and evaluation-facing question packages before release. Graph-object review checks macro nodes, textual and multimodal KCs, and reasoning edges; package review checks questions or thread seeds together with their target KCs, evidence, rubrics, forbidden claims, and linked assets or edges.

Graph object	N	Agr. (κ)	Valid	Rev.	Rem.
Macro nodes	98	0.92	95.9	3.1	1.0
Textual KC nodes	1286	0.91	95.3	3.3	1.4
MM KC nodes	469	0.89	92.8	5.0	2.2
Reasoning edges	1854	0.88	92.1	5.2	2.7

Table 3: Double-expert audit of graph objects. N is the pre-removal count; percentages split valid, revised, and removed items.

Package type	N	Agr. (κ)	Valid	Rev.	Rem.
Macro-KC	97	0.94	97.9	2.1	0.0
Challenge	242	0.89	94.1	4.2	1.7
MM Challenge	121	0.88	92.4	5.1	2.5
Thread Seed	188	0.86	91.8	5.5	2.7

Table 4: Double-expert audit of evaluation-facing packages. N is the pre-removal count; removed packages are excluded from runs.

Tables 3 and 4 show consistently high agreement and low removal rates across graph objects and packages. Revisions concentrate more on multimodal and reasoning-dependent items, but most audited items are either directly valid or corrected before release, indicating that the construction pipeline produces reliable graph and question artifacts after review.

4.2 Automatic Judge Validity

We compare automatic judge outputs with blind expert labels on approximately 20% of evaluation turns and repair tasks. The calibration covers the decisions used by our metrics, including KC coverage, macro completion, challenge outcomes, hallucination events and types, and repair success.

Table 5 shows strong agreement between automatic and human judgments across coverage, challenge, hallucination, and repair decisions. The pattern supports using the automatic judge for aggregate trajectory metrics, while keeping hallucination analyses event-based and type-aware rather than treating each event as an infallible label.

5 Experiments

5.1 Setup

We evaluate ten frontier models: Claude-Haiku, GPT-5 and GPT-5-mini (Singh et al., 2026), Gemini-3-Flash, Qwen3.6-Plus and Qwen3.6-Flash (Yang et al., 2025), Doubao-2-Pro and Doubao-2-Mini, Kimi-K2.5 (Team et al., 2026), and GLM-4.5v (V Team et al., 2025). All generations use temperature 1. To reduce sensitivity to

Judge task	Prec.	Rec.	F1 / Acc.
KC coverage	0.96	0.95	0.95
Macro completion	—	—	0.96
Challenge outcome	0.95	0.93	0.94
Text challenge outcome	0.96	0.94	0.95
MM challenge outcome	0.94	0.92	0.93
Hallucination event	0.94	0.92	0.93
Hallucination type	0.92	0.90	0.91
Repair success	—	—	0.93

Table 5: Human–judge calibration on core trajectory decisions. Accuracy is reported for single-label decisions.

stochastic decoding on the 12-paper set, we repeat the main-model runs and key ablations three times and report averaged table values. Appendix A.2 reports cost statistics, and Appendix A.4 gives paper-level bootstrap checks.

5.2 Metrics

Table 6 groups the main metrics into four aspects. *Knowledge coverage* measures whether a model covers target KCs in its first macro answer: KC Cov. is the target-KC coverage ratio, and Macro Full is the one-turn macro completion rate. *Challenge robustness* reports overall, text-only, and multimodal challenge failure. *Hallucination* reports the mean number of hallucination events per paper trajectory. *Cost* reports the average number of turns per paper and average response length per turn; Appendix Table 13 gives the full turn and response-length statistics. Repair metrics are computed from trajectory-linked repair tasks: Succ. is the task success rate before termination, and Avg Rounds is the number of repair follow-up turns used by the task.

5.3 Main Results

The results illustrate that the proposed metrics separate behaviors that would be conflated in static QA. Appendix A.5 reports the cross-domain results, whose overall patterns remain stable across the evaluated domains.

Finding 1: high coverage does not imply challenge robustness. Table 6 shows that high first-pass coverage does not guarantee resistance to misleading questions: Claude-Haiku-4.5 covers the paper most completely, while GPT-5-mini is strongest on challenge and hallucination metrics. The cost columns help interpret this divergence, as GPT-5 gives much shorter answers than GPT-5-mini but performs worse on trajectory-sensitive metrics,

Model	Knowledge Coverage		Challenge Robustness			Hallucination	Cost	
	KC Cov. ↑	Macro Full ↑	Chal. Fail ↓	Text Fail ↓	MM Fail ↓	Hallu Events ↓	Turns	Chars/Turn
<i>Closed-source models</i>								
Claude-Haiku-4.5	96.1	88.7	31.5	35.8	27.1	10.7	52.0	8385.6
GPT-5	88.0	68.7	33.2	35.8	30.5	10.1	60.8	832.2
GPT-5-mini	93.0	79.7	27.1	27.8	26.4	6.2	51.3	1858.6
Gemini-3-Flash	87.9	76.5	38.7	40.8	36.4	13.2	62.8	1924.3
Doubao-2-Pro	90.4	71.9	31.5	33.3	29.7	10.8	57.8	2168.0
Doubao-2-Mini	90.1	77.4	42.9	44.2	41.5	14.2	63.4	1790.2
Qwen3.6-Plus	87.9	74.4	37.8	39.2	36.4	13.2	63.9	1912.0
Qwen3.6-Flash	89.0	73.8	39.1	44.2	33.9	13.7	61.6	2179.0
<i>Open-source models</i>								
Kimi-K2.5	89.8	74.3	33.2	31.7	34.7	10.8	58.2	2210.2
GLM-4.5v	80.4	59.9	52.1	56.7	47.5	21.5	76.0	1736.1

Table 6: Main evaluation results. Rate columns are shown as percentages. Turns is the average number of dialogue turns per paper, and Chars/Turn is the average response length. Arrows indicate the preferred direction for score and error columns; cost columns are descriptive.

Model	Detail Repair		Hallucination Repair	
	Succ. ↑	Rounds ↓	Succ. ↑	Rounds ↓
<i>Closed-source models</i>				
Claude-Haiku-4.5	84.0	1.46	97.7	1.05
GPT-5	90.7	1.29	83.0	1.54
GPT-5-mini	97.1	1.13	74.6	1.55
Gemini-3-Flash	88.1	1.33	67.2	1.62
Doubao-2-Pro	92.3	1.24	75.5	1.50
Doubao-2-Mini	93.8	1.19	66.9	1.57
Qwen3.6-Plus	88.2	1.34	64.0	1.74
Qwen3.6-Flash	87.8	1.26	70.4	1.57
<i>Open-source models</i>				
Kimi-K2.5	91.5	1.28	81.4	1.40
GLM-4.5v	83.2	1.41	56.7	1.82

Table 7: Repair-task results. Succ. is shown as a percentage, and Rounds is the average number of repair follow-up turns.

suggesting that overly concise responses may omit evidence links or corrective details.

Finding 2: modality-conditioned failures reveal distinct weaknesses. The text/multimodal split shows that challenge robustness is not a single scalar ability: Qwen3.6-Flash is weaker on text traps, while Kimi-K2.5 shows the reverse pattern. Since text challenges stress unsupported scientific claims and multimodal challenges stress figure/table grounding, aggregate challenge failure alone would hide these differences.

Finding 3: repair metrics distinguish recoverable omissions from persistent hallucinations. Table 7 shows that recovering omitted content is generally easier than repairing hallucinations. A missing KC can often be recovered through a targeted follow-up, whereas an unsupported claim

	Contr. KC	False prem.	Over-claim	Wrong rel.	Total
GPT-5-mini	5	43	14	12	74
GPT-5	32	52	26	11	121
Claude-Haiku-4.5	39	47	21	21	128
Doubao-2-Pro	38	44	31	16	129
Kimi-K2.5	40	53	23	14	130
Gemini-3-Flash	39	48	42	29	158
Qwen3.6-Plus	49	52	36	22	159
Qwen3.6-Flash	53	51	37	23	164
Doubao-2-Mini	41	63	37	29	170
GLM-4.5v	107	74	48	29	258

Figure 5: Hallucination event counts by model after merging thread-specific variants into the four benchmark failure categories. Each horizontal bar is stacked by event type.

may persist after the model has committed to a false premise or overclaim; repair metrics therefore capture behavior that first-turn coverage cannot.

Finding 4: event categories expose graph-local errors. Figure 5 shows that many hallucinations are not generic factual mistakes, but graph-local errors such as accepting a false premise, contradicting a KC, or extending a paper claim beyond its evidence. Event-level tracking therefore explains how an answer deviates from the paper graph, not just whether it is wrong.

Finding 5: hallucinations can accumulate across turns. Figure 6 shows that GPT-5’s hallucination events fluctuate but rise overall as the dialogue proceeds. This suggests that earlier unsupported claims can affect later turns, so long trajectories should be evaluated not only by final accuracy but also by how errors are introduced, repaired, or carried forward.

Model	Knowledge Coverage		Challenge Robustness			Hallucination	Cost	
	KC Cov. $\uparrow$	Macro Full $\uparrow$	Chal. Fail $\downarrow$	Text Fail $\downarrow$	MM Fail $\downarrow$	Hallu Events $\downarrow$	Turns	Chars/Turn
GPT-5	88.0	68.7	33.2	35.8	30.5	10.1	60.8	832.2
w/o Repair	89.3	69.0	40.3	44.2	36.4	9.8	38.2	870.5
w/o MT	88.5	68.6	40.3	40.8	39.8	—	—	—
w/o Graph	—	—	13.3	14.2	12.5	4.9	28.0	648.5
GPT-5-mini	93.0	79.7	27.1	27.8	26.4	6.2	51.3	1858.6
w/o Repair	93.0	79.3	32.4	34.2	30.5	7.8	38.0	1841.8
w/o MT	91.5	76.2	32.4	27.5	37.3	—	—	—
w/o Graph	—	—	11.2	10.0	12.5	5.2	28.0	1318.1
Doubao-2-Pro	90.4	71.9	31.5	33.3	29.7	10.8	57.8	2168.0
w/o Repair	91.0	75.3	40.8	46.7	34.7	10.5	38.0	2337.9
w/o MT	90.2	70.8	45.4	52.5	38.1	—	—	—
w/o Graph	—	—	16.7	18.3	15.0	9.2	28.0	1658.3
Doubao-2-Mini	90.1	77.4	42.9	44.2	41.5	14.2	63.4	1790.2
w/o Repair	89.9	76.5	46.6	49.2	44.1	11.6	38.0	2005.3
w/o MT	90.7	73.3	46.2	46.7	45.8	—	—	—
w/o Graph	—	—	20.8	26.7	15.0	11.1	28.0	1517.9

Table 8: Ablation of graph guidance, multi-turn interaction, and repair. Full uses the complete graph-guided dialogue; w/o Repair removes repair turns; w/o MT removes state-conditioned follow-up and repair interactions; and w/o Graph removes graph guidance from benchmark construction and evaluation. Dashes denote metrics that are undefined or incomparable under the corresponding setting.

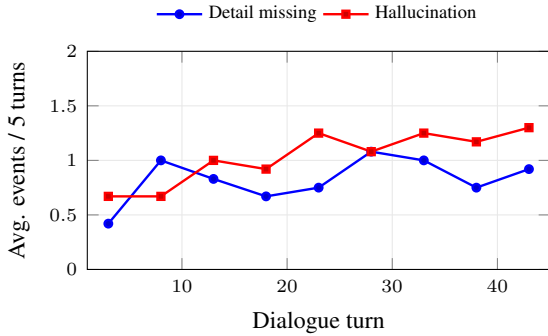


Figure 6: GPT-5 repair-event trend across dialogue turns. Values are averaged over five-turn bins and twelve paper trajectories.

Method	KC Cov. $\uparrow$	Macro Full $\uparrow$	Chal. Fail $\downarrow$
GPT-5-mini	93.0	79.7	27.1
PaperQA2	81.2	58.0	32.4
Δ	-11.7	-21.7	+5.2

Table 9: Comparison between the GPT-5-mini-based PaperQA2 pipeline and direct GPT-5-mini.

Finding 6: scientific RAG does not replace trajectory-level paper reading. PaperQA2 (Lála et al., 2023) supports text input only. With the same GPT-5-mini backbone, Table 9 shows lower knowledge coverage and macro completion and higher challenge failure than direct GPT-5-mini. This indicates that retrieval and synthesis alone do not ensure comprehen-

sive paper coverage or challenge robustness, and therefore do not subsume the multimodal, stateful, and repair-oriented abilities evaluated by PAPERGRAPH-BENCH.

5.4 Ablation

We study three interventions. The w/o Repair setting removes repair turns, w/o MT removes all state-conditioned follow-up and repair interactions, and w/o Graph removes the graph from both benchmark construction and evaluation while retaining dialogue history and required visual inputs.

Repair ablation. Removing repair turns substantially shortens trajectories, so hallucination events would normally have fewer opportunities to occur. However, Table 8 shows that hallucination counts remain close to the full setting and even increase for GPT-5-mini. Challenge failure also rises consistently under w/o Repair. This indicates that repair turns do more than resolve local gaps: they add corrective context that helps later answers avoid unsupported claims and misleading premises.

Multi-turn ablation. Removing the multi-turn mechanism raises challenge failure for every ablated model, with the largest degradation on Doubao-2-Pro. First-pass coverage remains comparatively stable, which suggests that the effect is not simply caused by easier or harder macro questions. Instead, the full protocol gives mod-

els chances to summarize prior evidence, observe correction patterns, and reuse paper-specific constraints when facing later challenges. The result provides evidence that models can use experience from earlier turns within a trajectory, so paper reading should be evaluated as a stateful process rather than as independent questions.

Graph ablation. Because graph guidance shapes benchmark construction, Full and w/o Graph use independently generated question packages; the comparison therefore measures the graph’s benchmark-level contribution rather than the effect of showing a graph to the evaluated model. Table 8 shows that removing the graph makes challenges consistently easier and compresses model differences, especially on multimodal items. Thus, graph-linked KCs, evidence boundaries, and verified relations enable more difficult scope- and relation-sensitive questions and improve the benchmark’s discriminative power. KC coverage and macro completion are undefined without graph-aligned targets and are marked as unavailable rather than zero.

6 Conclusion

We presented PAPERGRAPH-BENCH, a graph-guided benchmark framework for evaluating multimodal, multi-turn scientific-paper reading. By representing papers with macro structure, evidence-grounded KCs, verified reasoning edges, and stateful evaluation trajectories, the benchmark makes coverage gaps, misleading-premise failures, hallucination events, and repair behavior visible in a unified record. Experiments show that coverage, robustness, modality-specific failures, hallucination profiles, and repair success capture distinct aspects of paper understanding, supporting graph-state trajectories as a diagnostic format for future scientific-paper evaluation.

Limitations

PAPERGRAPH-BENCH has three main limitations. (1) Its primary evaluation is a compact, depth-oriented prototype built from NLP papers. Our 12-paper extension to three additional domains provides preliminary evidence that the pipeline and overall behavioral patterns transfer, but broader claims require larger and more diverse paper sets. (2) Construction and judging are LLM-assisted and human-audited, so residual graph, question, or judge errors may remain; reported model numbers should be read as frozen-run diagnostics rather than

a permanent capability ranking. (3) The benchmark evaluates paper-grounded reading, not full scientific verification: it does not rerun experiments, inspect code or raw data, or validate claims against evidence outside the paper.

Ethics Statement

The benchmark is built from publicly available scientific papers, and any release should preserve citation information and respect the copyright and redistribution policies of the original papers and figures. Because the graph, questions, and judge outputs are partly LLM-generated, the benchmark is intended for aggregate diagnostic analysis rather than authoritative judgment of individual answers. Some challenge questions contain false premises or unsupported relations by design; they are used to test whether models reject misleading scientific claims, not to promote deceptive or unsupported scientific statements.

Acknowledgments

This work is supported by the New Generation Artificial Intelligence-National Science and Technology Major Project (No. 2025ZD0123602)

References

- Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. 2024. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2024, Bangkok, Thailand, August 11-16, 2024, pages 7421–7454.
- Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 3639–3664.
- Kedi Chen, Qin Chen, Jie Zhou, He Yishen, and Liang He. 2024. Diahalu: A dialogue-level hallucination evaluation benchmark for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9057–9079.
- Shiqi Chen, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. 2023. FELM: benchmarking factuality evaluation of large

- language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Xi Chen, Mao Mao, Shuo Li, and Haotian Shangguan. 2025. Debate-feedback: A multi-agent framework for efficient legal judgment prediction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 2: Short Papers, Albuquerque, New Mexico, April 29 - May 4, 2025*, pages 462–470.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, pages 8359–8388.
- Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, Yue Zhang, Yubo Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. 2026. [Paddleocr-vl-1.5: Towards a multi-task 0.9b vlm for robust in-the-wild document parsing](#). Preprint, arXiv:2601.21957.
- Jisheng Dang, Shengjun Deng, Haochen Chang, Teng Wang, Bimei Wang, Shude Wang, Nannan Zhu, Guo Niu, Jingwen Zhao, and Jizhao Liu. 2025. Hallucination reduction in video-language models via hierarchical multimodal consistency. In *Proceedings of the Thirty-Fourth International Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pages 9167–9175.
- Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610.
- Chengfeng Dou, Ying Zhang, Yanyuan Chen, Zhi Jin, Wenpin Jiao, Haiyan Zhao, Yongqiang Zhao, Zhenwei Tao, and Yun Huang. 2024. Detection, diagnosis, and explanation: A benchmark for Chinese medical hallucination evaluation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4784–4794.
- Dongyang Fan, Sebastien Delsad, Nicolas Flammarion, and Maksym Andriushchenko. 2026. [Halluhard: A hard multi-turn hallucination benchmark](#). Preprint, arXiv:2602.01031.
- Ting-Yao Hsu, C Lee Giles, and Ting-Hao Huang. 2021. SciCap: Generating captions for scientific figures. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3258–3264.
- Chaoya Jiang, Haiyang Xu, Mengfan Dong, Jiaxing Chen, Wei Ye, Ming Yan, Qinghao Ye, Ji Zhang, Fei Huang, and Shikun Zhang. 2024. Hallucination augmented contrastive learning for multimodal large language model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 27026–27036.
- Jakub Lála, Odhran O’Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G Rodrigues, and Andrew D White. 2023. Paperqa: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*.
- Eric Lehman, Jay DeYoung, Regina Barzilay, and Byron C. Wallace. 2019. Inferring which medical treatments work from reports of clinical trials. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3705–3717.
- Chuhan Li, Ziyao Shangguan, Yilun Zhao, Deyuan Li, Yixin Liu, and Arman Cohan. 2024. M3sciqa: A multi-modal multi-document scientific QA benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 15419–15446.
- Jinnan Li, Jinzhe Li, Yue Wang, Yi Chang, and Yuan Wu. 2025. Structflowbench: A structured flow benchmark for multi-turn instruction following. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9322–9341.
- Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252.
- Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yungang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. 2024. MMLONGBENCH-DOC: benchmarking long-context document understanding with visualizations. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark

- for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279.
- Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. 2022. Infographicvqa. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022*, pages 2582–2591.
- Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. Docvqa: A dataset for VQA on document images. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, pages 2199–2208.
- Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pages 1516–1525.
- Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100.
- Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878.
- Kaipeng Niu, Danni Xu, Bingjian Yang, Wenxuan Liu, and Zheng Wang. 2025. Pioneering explainable video fact-checking with a new dataset and multi-role multimodal model approach. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 28276–28283.
- Changwon Ok, Eunkeyeong Lee, and Dongsuk Oh. 2025. Synthetic paths to integral truth: Mitigating hallucinations caused by confirmation bias with synthetic data. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 5168–5180.
- Yeji Park, Deokyeon Lee, Junsuk Choe, and Buru Chang. 2025. Convis: Contrastive decoding with hallucination visualization for mitigating hallucinations in multimodal large language models. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 6434–6442.
- Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2024. SPIQA: A dataset for multimodal question answering on scientific papers. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Chengwei Qin, Wenxuan Zhou, Karthik Abinav Sankararaman, Nanshu Wang, Tengyu Xu, Alexander Radovic, Eryk Helenowski, Arya Talebzadeh, Aditya Tayade, Sinong Wang, Shafiq Joty, Han Fang, and Hao Ma. 2025. Learning auxiliary tasks improves reference-free hallucination detection in open-domain long-form generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 1173–1182.
- Alex Robertson, Huizhi Liang, Mahbub Gani, Rohit Kumar, and Srijith Rajamohan. 2026. Kghalubench: A knowledge graph-based hallucination benchmark for evaluating the breadth and depth of llm knowledge. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 3975–3989.
- Hannah Sansford, Nicholas Richardson, Hermina Petric Maretic, and Juba Nait Saada. 2024. Grapheval: A knowledge-graph based LLM hallucination evaluation framework. In *Proceedings of the Workshop on Knowledge-infused Learning co-located with 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Barcelona, Spain, August 26, 2024*, pages 20–31.
- Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 467 others. 2026. [Openai gpt-5 system card](#). Preprint, arXiv:2601.03267.
- Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Cheng Chen, Guanduo Chen, Huarong Chen, Jia Chen, Jianlong Chen, Jun Chen, Kefan Chen, Liang Chen, Ruijue Chen, Xinhao Chen, and 318 others. 2026. [Kimi k2.5: Visual agentic intelligence](#). Preprint, arXiv:2602.02276.
- Yang Tian, Zheng Lu, Mingqi Gao, Zheng Liu, and Bo Zhao. 2025. MMCR: Benchmarking cross-source reasoning in scientific papers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 488–497.

- V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Wei Han Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, and 74 others. 2025. GLM-4.5V and GLM-4.1V-Thinking: Towards Versatile Multimodal Reasoning with Scalable Reinforcement Learning. *arXiv e-prints*, page arXiv:2507.01006.
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550.
- Luanbo Wan and Weizhi Ma. 2025. *Storybench: A dynamic benchmark for evaluating long-term memory with multi turns*. Preprint, arXiv:2506.13356.
- Chengye Wang, Yifei Shen, Zexi Kuang, Arman Cohan, and Yilun Zhao. 2025a. Sciver: Evaluating foundation models for multimodal scientific claim verification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 8562–8579.
- Sudong Wang, Yunjian Zhang, Yao Zhu, Enci Liu, Jianing Li, Yanwei Liu, and Xiangyang Ji. 2025b. SHIFT: smoothing hallucinations by information flow tuning for multimodal large language models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pages 3639–3649.
- Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Hao-tian Liu, Sadhika Malladi, Alexis Chevalier, Sanjeev Arora, and Danqi Chen. 2024. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. Long-form factuality in large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. *Qwen3 technical report*. Preprint, arXiv:2505.09388.
- Shengbin Yue, Siyuan Wang, Wei Chen, Xuanjing Huang, and Zhongyu Wei. 2025. Synergistic multi-agent framework with trajectory learning for knowledge-intensive tasks. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 25796–25804.
- Xuanliang Zhang, Dingzirui Wang, Baoxin Wang, Longxu Dou, Xinyuan Lu, Keyan Xu, Dayong Wu, and Qingfu Zhu. 2025. SCITAT: A question answering benchmark for scientific tables and text covering diverse reasoning types. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 3859–3881.
- Zhehao Zhang, Jiaao Chen, and Diyi Yang. 2024. DARG: dynamic evaluation of large language models via adaptive reasoning graph. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Li Zheng, Hao Fei, Ting Dai, Zuquan Peng, Fei Li, Huisheng Ma, Chong Teng, and Donghong Ji. 2025. Multi-granular multimodal clue fusion for meme understanding. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 26057–26065.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

A Experimental Supplement

A.1 Model Details

Table 10 lists the evaluated model snapshots and their public release or knowledge-cutoff dates. These dates help contextualize the source-paper selection in Table 16, where we prioritize recent papers to reduce the chance of model memorization.

Model	Release Date	Knowledge Cutoff
Claude-Haiku-4.5	Oct. 2025	Jan.–Feb. 2025
GPT-5	Aug. 2025	Sep. 2024
GPT-5-mini	Aug. 2025	May 2024
Gemini-3-Flash	Dec. 2025	Jan. 2025
Qwen3.6-Plus	Apr. 2026	Apr. 2025
Qwen3.6-Flash	Apr. 2026	Apr. 2025
Doubao-2-Pro	Feb. 2026	–
Doubao-2-Mini	Feb. 2026	–
Kimi-K2.5	Jan. 2026	Jan. 2025
GLM-4.5v	Aug. 2025	Feb. 2025

Table 10: Release dates and knowledge cutoffs of the evaluated model snapshots.

A.2 Cost Details

Table 13 reports the frozen-run dialogue cost in turns and response length. Figure 7 summarizes available model-evaluation billing logs, and Table 11 reports construction and judge costs over the 12-paper prototype benchmark.

Figure 7 plots the available model-evaluation spending logs. When a model has multiple billing components, we sum them before plotting. Table 11 reports wall-clock time and RMB-denominated construction or judge costs over the 12-paper prototype benchmark.

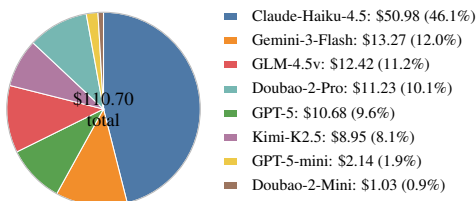


Figure 7: Model-evaluation spending for runs with available billing logs. Values are in USD and sum to \$110.70.

Item	Total	Per paper avg.
Data construction time	19h00min	1h35min
Evaluation time	14h36min	1h13min
Data construction cost	RMB 43.00	RMB 3.58
Evaluation judge cost	RMB 84.00	RMB 7.00

Table 11: Average construction and evaluation cost per source paper. Totals are computed over 12 papers.

A.3 Evaluation Metrics

Let $\mathcal{T}$ be all turns, $\mathcal{M}$ macro turns, $\mathcal{C}$ challenge turns, $\mathcal{C}_{txt}$ text challenge turns, and $\mathcal{C}_{mm}$ multimodal challenge turns. For a turn t , let $\text{tar}(t)$ be target KCs, $\text{cov}(t)$ covered KCs, and H_t hallucination events. Table 15 summarizes the main metrics.

A.4 Significance Test

We test the robustness of the main best-model findings with clustered bootstrap over papers. For each model and paper, we first average repeated runs into a paper-level metric, select the raw-mean best model for each metric, and estimate its 90% confidence interval by resampling the 12 papers with replacement for 20,000 trials. We also compare the best model with each other model using paired paper-level differences; for lower-is-better metrics, the difference is direction-normalized so positive values always favor the best model.

Metric	Best model	Best mean	90% CI	Sig.
KC Cov.	ClaudeHaiku4.5	96.1	[94.2, 97.8]	9/9
M.F.	ClaudeHaiku4.5	88.7	[83.4, 93.7]	9/9
Chal. Fail	GPT-5-mini	27.0	[23.2, 31.2]	8/9
Text Fail	GPT-5-mini	27.2	[21.0, 35.0]	7/9
MM Fail	GPT-5-mini	27.1	[21.3, 32.5]	6/9
H.E.	GPT-5-mini	6.17	[5.00, 7.33]	9/9

Table 12: Best-metric clustered bootstrap summary. Sig. reports how many best-vs-other paired comparisons have a 90% effect-size CI entirely above zero.

Table 12 shows that coverage, macro completion, and hallucination-event conclusions are stable under paper-level resampling. GPT-5-mini remains the best challenge model overall, but multimodal challenge failure is more sensitive to paper heterogeneity, indicating that figure/table-grounded traps vary more across source papers.

A.5 Cross-Domain Evaluation

To examine whether the evaluation behavior is specific to NLP papers, we apply the same graph construction and trajectory evaluation protocol to 12

Model	Papers	Turns	Avg. chars/turn	Median	Min	Max
Claude-Haiku-4.5	12	624	8385.6	7607.0	690	30557
Kimi-K2.5	12	698	2210.2	2053.0	144	6453
Qwen3.6-Flash	12	739	2179.0	2038.0	103	5359
Doubao-2-Pro	12	694	2168.0	2023.5	134	6470
Gemini-3-Flash	12	754	1924.3	1961.0	75	4232
Qwen3.6-Plus	12	767	1912.0	1742.0	87	5974
GPT-5-mini	12	616	1858.6	1569.5	51	7607
Doubao-2-Mini	12	761	1790.2	1539.0	181	6303
GLM-4.5v	12	912	1736.1	1540.5	131	5589
GPT-5	12	730	832.2	647.5	6	4239

Table 13: Response-length statistics for the frozen evaluation runs. Lengths are measured in characters per model answer.

Model	Domain	Knowledge Coverage		Challenge Robustness			Hallucination	Cost	
		KC Cov. ↑	Macro Full ↑	Chal. Fail ↓	Text Fail ↓	MM Fail ↓	Hallu Events ↓	Turns	Chars/Turn
GPT-5	NLP	88.0	68.7	33.2	35.8	30.5	10.1	60.8	832.2
GPT-5	Medical/Public Health	93.2	78.1	32.5	20.0	45.0	12.0	60.0	733.4
GPT-5	Finance	91.6	79.5	33.8	35.0	32.5	9.8	56.5	886.3
GPT-5	Law	83.0	62.5	28.0	32.5	22.9	7.8	54.0	1054.2
GPT-5-mini	NLP	93.0	79.7	27.1	27.8	26.4	6.2	51.3	1858.6
GPT-5-mini	Medical/Public Health	91.6	84.4	30.0	22.5	37.5	11.5	57.8	1580.0
GPT-5-mini	Finance	92.4	93.8	26.2	30.0	22.5	6.8	49.5	2214.8
GPT-5-mini	Law	90.2	75.0	22.7	22.5	22.9	7.2	53.0	2077.0
Doubao-2-Pro	NLP	90.4	71.9	31.5	33.3	29.7	10.8	57.8	2168.0
Doubao-2-Pro	Medical/Public Health	89.9	84.4	31.2	30.0	32.5	11.0	56.2	2172.6
Doubao-2-Pro	Finance	94.9	85.1	32.5	25.0	40.0	10.5	54.5	2384.2
Doubao-2-Pro	Law	86.7	71.9	32.0	42.5	20.0	9.8	58.0	2755.1
Doubao-2-Mini	NLP	90.1	77.4	42.9	44.2	41.5	14.2	63.4	1790.2
Doubao-2-Mini	Medical/Public Health	86.6	78.1	45.0	30.0	60.0	16.0	67.5	1778.7
Doubao-2-Mini	Finance	94.0	82.3	41.2	40.0	42.5	14.5	64.2	1998.1
Doubao-2-Mini	Law	86.7	71.9	41.3	42.5	40.0	12.5	63.8	2203.5

Table 14: Cross-domain evaluation results. The NLP rows reproduce the corresponding primary-benchmark results for direct comparison; each additional domain contains four papers. Rate columns are shown as percentages, arrows indicate the preferred direction for score and error columns, and cost columns are descriptive.

additional papers, with four papers each from Medical and Public Health, Finance, and Law. We evaluate four representative models and retain the NLP results in Table 14 for direct comparison.

The principal behavioral patterns remain broadly stable across domains: relative differences in macro completion and challenge robustness persist, and high coverage does not uniformly imply stronger resistance to misleading premises. Domain-level variation is nevertheless visible across text and multimodal challenges, indicating that different scientific evidence structures stress different aspects of paper reading. These results provide preliminary evidence that the framework transfers beyond NLP while leaving broader generalization to larger domain-specific collections.

A.6 Benchmark Source Papers

Table 16 lists the twelve manually selected papers used to build the primary NLP benchmark. We select papers with at least five figures or tables

and a length of 8–12 pages, while balancing topics and evidence types to keep the generated dataset relatively even across construction and evaluation units. We also prioritize papers from the most recent year available: the current pool covers 2024–2025 and is dominated by 2025 papers. Compared with the model knowledge cutoffs in Table 10, this recency reduces the chance that models answer from memorized paper-specific content rather than the provided paper context. These papers provide the source documents from which paper graphs, evaluation packages, and trajectory instances are constructed.

B Graph Construction Details

For each raw PDF, we first use PaddleOCR-VL-1.5 (Cui et al., 2026) to parse the paper into Markdown files and extracted image assets. The cleaned Markdown is then used as the source corpus for paper-graph construction, multimodal asset group-

Metric	Definition	Interpretation
Total turns	$ \mathcal{T} $	Dialogue cost and evaluation length.
One-shot macro completion	$ \{t \in \mathcal{M} : \text{tar}(t) \subseteq \text{cov}(t)\} / \mathcal{M} $	Fraction of macro questions answered with all target KCs in one turn.
One-shot KC coverage	$\sum_{t \in \mathcal{M}} \text{cov}(t) / \sum_{t \in \mathcal{M}} \text{tar}(t) $	Overall target-KC coverage in first macro answers.
Challenge failure	$ \{t \in \mathcal{C} : \text{failed}(t)\} / \mathcal{C} $	Rate of accepting misleading challenge premises or relations.
Text challenge failure	$ \{t \in \mathcal{C}_{txt} : \text{failed}(t)\} / \mathcal{C}_{txt} $	Failure rate on text-grounded challenges.
Multimodal challenge failure	$ \{t \in \mathcal{C}_{mm} : \text{failed}(t)\} / \mathcal{C}_{mm} $	Failure rate on figure/table-grounded challenges.
Hallucination-turn ratio	$ \{t \in \mathcal{T} : H_t \neq \emptyset\} / \mathcal{T} $	Share of turns that produce at least one hallucination event.
Hallucination type count	$\text{count}(h.\text{type})$ for $h \in \bigcup_t H_t$	Distribution over false-premise acceptance, overclaiming, wrong relation, thread hallucination, and other event types.

Table 15: Primary metrics computed from `dialogue_trajectory.json`.

ID	Paper	Venue / Year	Primary signal
P1	ConVis (Park et al., 2025)	AAAI 2025	Multimodal hallucination mitigation
P2	Hierarchical Multimodal Consistency (Dang et al., 2025)	IJCAI 2025	Video-language hallucination reduction
P3	Multi-Granular Multimodal Clue Fusion (Zheng et al., 2025)	AAAI 2025	Multimodal clue grounding
P4	Synergistic Multi-Agent Framework (Yue et al., 2025)	AAAI 2025	Trajectory learning for knowledge tasks
P5	Debate-Feedback (Chen et al., 2025)	NAACL 2025	Multi-agent legal reasoning
P6	Hallucination Augmented Contrastive Learning (Jiang et al., 2024)	CVPR 2024	Multimodal hallucination training
P7	Auxiliary Tasks for Reference-Free Hallucination Detection (Qin et al., 2025)	ACL 2025	Long-form hallucination detection
P8	SHIFT (Wang et al., 2025b)	ICCV 2025	Information-flow hallucination smoothing
P9	Chinese Medical Hallucination Evaluation (Dou et al., 2024)	LREC-COLING 2024	Domain hallucination benchmark
P10	Explainable Video Fact-Checking (Niu et al., 2025)	AAAI 2025	Multimodal fact-checking
P11	Synthetic Paths to Integral Truth (Ok et al., 2025)	COLING 2025	Confirmation-bias hallucination mitigation
P12	StoryBench (Wan and Ma, 2025)	CoRR 2025	Multi-turn long-term memory evaluation

Table 16: Manually selected source papers used for constructing the primary NLP benchmark.

ing, question generation, and evaluation context building.

B.1 Macro Spine Construction

The macro spine is produced from parsed paper sections with a target count of 8 and an allowed extraction range of 6–12. The extractor is not forced to copy section titles or follow a fixed IMRaD taxonomy. Figure 9 in Section F shows a real macro object from the StoryBench paper graph.

B.2 KC Object Schema

KCs are extracted from textual units and multimodal asset explanations. A valid KC must be a concrete claim rather than a keyword, must be supported by the provided evidence, and must be

usable for answer judging or claim verification. The textual extractor allows claims about problems, methods, mechanisms, datasets, experiments, ablations, results, conclusions, limitations, and related paper-specific content. The multimodal extractor uses a narrower inventory for asset-grounded claims, such as table result, table comparison, chart trend, visual mechanism, visual pipeline, visual component, and multimodal limitation.

Each accepted textual KC stores a claim, evidence paragraph ids, macro id, type, importance, claim strength, scope, and related terms. Multimodal KCs additionally store the asset id, asset type, evidence basis, and modality class. For answer judging, the KC bank later attaches acceptable

variants and forbidden claims. Figure 10 in Section F shows two real KC objects with only the most audit-relevant fields retained.

B.3 Reasoning Edge Verification

Reasoning edge candidates are proposed at four scopes: local unit, within macro, cross macro, and thread level. Each candidate includes a source KC, target KC, relation, evidence packet, and natural-language rationale. The verifier then checks whether the relation and direction are supported by the packet. Multimodal packets may contain prepared virtual units derived from figures or tables; for these cases the verifier uses the caption, asset summary, evidence basis, and supported multimodal KCs as evidence. Table 17 summarizes the verifier’s decision space.

The allowed relation labels are *part_of*, *defines*, *motivates*, *solves*, *explains*, *supports*, *contrasts_with*, and *limits*. Only edges marked valid or correctable by relation relabeling are inserted into the final graph. If the verifier changes the relation label, the corrected label is stored as the verified relation. All accepted edges retain the verifier confidence and reason, which are later available for challenge planning and manual inspection. Figure 11 in Section F shows two verified edges from the same graph.

B.4 Active KC Scoring

KC scoring is used only to prioritize which graph nodes become active targets for macro questions. It is not used as a factual label and does not remove lower-scoring KCs from the graph. The implementation combines five normalized signals:

$$r(k) = 0.30 c_M(k) + 0.25 q_{ev}(k) + 0.20 s(k) + 0.15 q_{ask}(k) + 0.10 c_G(k). \quad (6)$$

Here c_M is macro centrality from the macro spine, q_{ev} is evidence quality, s is claim specificity, q_{ask} is questionability, and c_G is graph connectivity. Evidence quality is computed from claim–evidence embedding similarity and normalized to $[0, 1]$. Specificity and questionability are scored by an LLM on a 1–5 rubric and linearly normalized to $[0, 1]$. Macro centrality and graph connectivity are graph-derived scores.

The default active-KC policy selects up to 30 KCs per paper, requires at least two active KCs per macro and at least three for critical macros when available, and uses a default score threshold of 0.65.

The policy first adds high-scoring KCs, then fills macro coverage minima, and finally truncates while preserving required per-macro coverage. These hyperparameters make the main questions dense enough to measure partial coverage while keeping each macro question judgeable.

B.5 Multimodal Core Coverage

Figures and tables are treated as graph evidence rather than as separate benchmark material. The multimodal pipeline groups assets, aligns them to nearby sections and macro nodes, generates asset explanations, and extracts at most four KCs per asset. It also records possible misreadings, but those misreadings are used as traps for challenge generation rather than as supported KCs.

The current release uses textual KCs as the default active targets for macro questions and reserves multimodal KCs for multimodal challenge questions, hallucination follow-ups, repair turns, and graph-state updates. This is a design choice for comparability: macro questions should assess whether a model understands the paper’s main argument, while multimodal challenges isolate errors that arise from interpreting figures or tables. The graph still stores multimodal KCs as first-class nodes with asset ids, evidence bases, and reasoning edges.

The implementation also supports an alternative setting in which multimodal KCs are eligible for active-KC selection. In that setting, asset-grounded KCs can become direct targets of macro coverage, but the resulting score is more sensitive to the quality of figure parsing and MLLM asset descriptions. We therefore report the text-default setting as the main benchmark configuration and keep multimodal-in-core coverage as an analysis option.

C Question Generation Details

C.1 Question Package Construction

Evaluation packages are generated from graph targets and later consumed by the scheduler. Figure 12 in Section F shows one macro question and one challenge question from the StoryBench graph.

C.2 Challenge Question Generation Loop

Challenge questions are generated with two validation loops. First, after a candidate question $q_{p,a}$ is generated from challenge plan p at attempt a , an LLM meta-judge checks whether it is natural, hides internal labels and ids, follows

Decision	Meaning
Valid	Relation, direction, and evidence are supported.
Invalid	Relation is unsupported or merely topical.
Insufficient context	Packet lacks enough evidence.
Wrong direction	The reverse edge may be valid, but the proposed direction is not.
Wrong relation label	The edge is real, but the label should be changed.

Table 17: Verifier decisions for reasoning edge candidates.

the intended failure mode, and can be judged from the paper evidence. If the question is unusable, the system records the reason and revision guidance as feedback $f_{p,a+1}$ and regenerates from the same plan. The outer loop allows at most $A = 3$ attempts per plan by default through CHALLENGE_MAX_ATTEMPTS_PER_PLAN; the generator also has a two-try internal repair for malformed JSON or exposed internal labels.

Second, usable questions are tested by solver trials. By default, $m = 3$ solvers answer each question using the paper context, and a judge checks whether each answer matches the intended failure mode. Let $z_i(q) = 1$ if solver i is trapped by q , and 0 otherwise. The filtering rule is:

$$s(q) = \sum_{i=1}^m z_i(q), \quad (7)$$

$$\text{accept}(q) = \mathbb{I}[s(q) > 0].$$

If no solver is trapped, a second meta-judge decides whether the plan itself is too easy or only the surface question is too easy. Let $e = e(p, q)$ denote the plan-easiness decision:

$$\begin{aligned} s(q) > 0 &\Rightarrow \text{accept}, \\ s(q) = 0, e = 1 &\Rightarrow \text{blacklist}, \\ s(q) = 0, e = 0 &\Rightarrow \text{regenerate}. \end{aligned} \quad (8)$$

Here $e = 1$ means that the meta-judge diagnoses the plan as inherently too easy. A blacklisted plan is removed from the pool; otherwise the next attempt reuses the same plan with the easiness judge’s revision guidance. If every solver answer is judged wrong, the accepted question is retained but marked for human review.

D Evaluation Protocol Details

D.1 Turn Types

Table 18 summarizes the trajectory scheduler’s turn types, their triggers, and the state fields updated after judging.

D.2 Trajectory Records and Judge Outputs

The evaluation trajectory is the primary record for experiments. Each turn stores the issued question, target KCs, model answer, judge output, state update, and next action. Figures 13 and 14 in Section F show a realized challenge turn and its judge output.

E Audit Supplement

E.1 Audit Interface

Figure 8 shows the visual audit platform used to inspect graph objects, evaluation packages, trajectory records, and judge-calibration samples.

We did not log item-level audit time in detail; with two annotators, the full audit for this prototype was completed within four days.

F Example Instances

This section collects representative records referenced by the graph construction, question generation, and evaluation protocol supplements. The examples are real outputs from our construction or evaluation artifacts, with only nonessential fields omitted for readability.

F.1 Graph Object Instances

Figure 9 shows a macro node from the StoryBench graph, including its role, evidence sections, reader-facing question, and graph links. Figure 10 then illustrates the compact schema used for textual and multimodal KCs, while Figure 11 shows verified reasoning edges that preserve relation labels, direction, evidence, and verifier confidence.

F.2 Question and Evaluation Instances

Figure 12 shows the fields retained in generated evaluation packages before scheduling. Figures 13 and 14 show a realized challenge turn and the corresponding judge output, making explicit how coverage, hallucination events, and next actions are stored.

Turn type	Trigger	Main target	State update
Macro	Start of each macro stage	First-pass coverage of active KCs	Covered and missing KCs; macro progress
Detail follow-up	Missing target KCs	Recover omitted paper content	Resolved or persistent coverage gap
Hallucination repair	Hallucination event or accepted misleading premise	Retraction or correction of unsupported content	Resolved, persistent, or exhausted event
Thread	Ready reasoning seed	Cross-KC reasoning over premise, evidence, bridge, or review roles	Thread step progress and bridge success
Challenge	Macro- or thread-grounded challenge plan	False premise, overclaim, or wrong relation	Challenge pass/fail and linked hallucination event
Review	End of the trajectory	Consistency after prior turns	Regression, consistency, and late repair tasks

Table 18: Evaluation turn types and their state updates.

G Prompt Templates

Figures 15–21 show the full original prompts used by the main construction and evaluation pipeline. Repeated variants, such as thread-specific or modality-specific judges with the same output contract, are omitted for brevity.

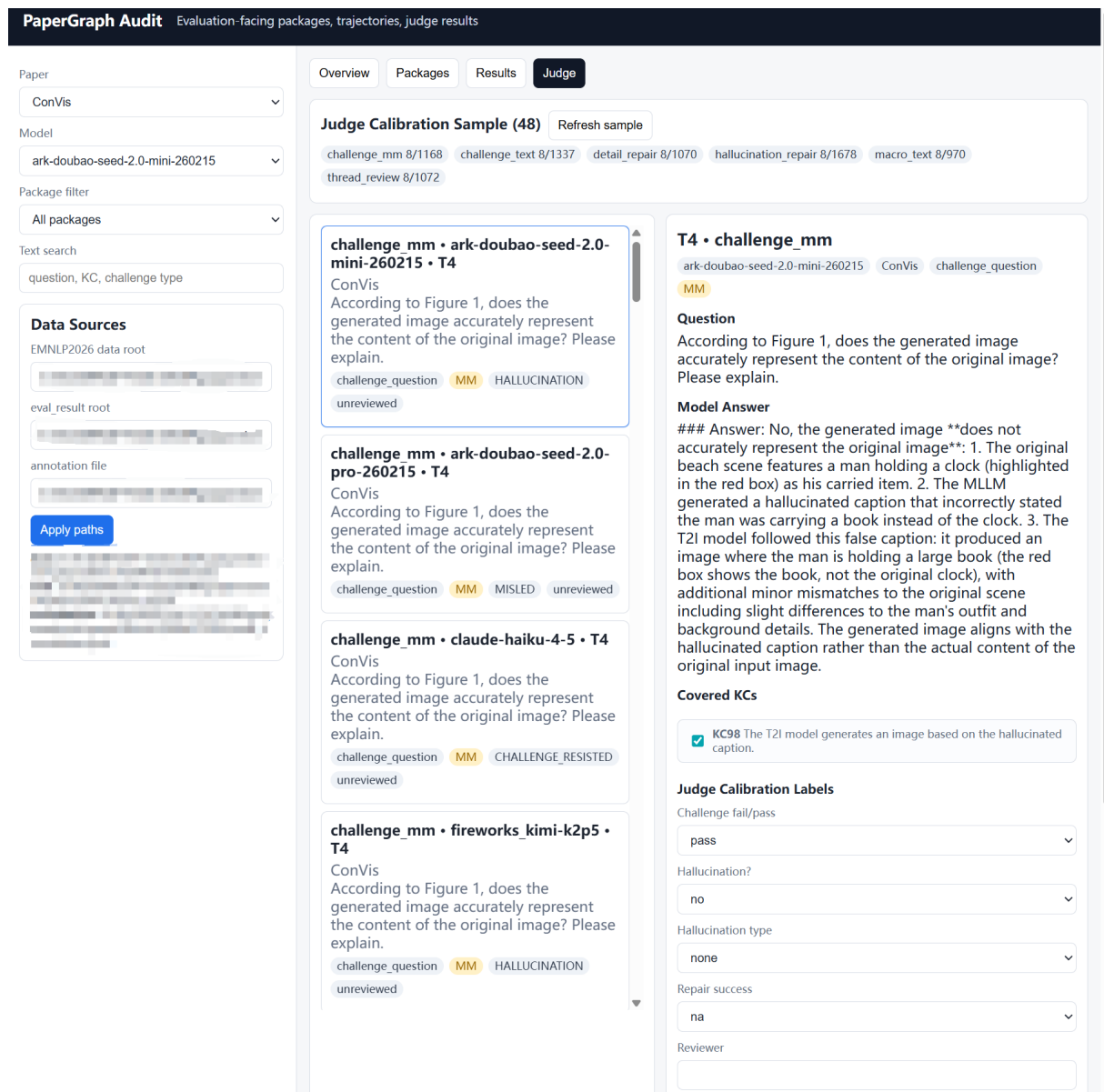


Figure 8: Visual audit platform for reviewing evaluation-facing packages, trajectories, and judge results.

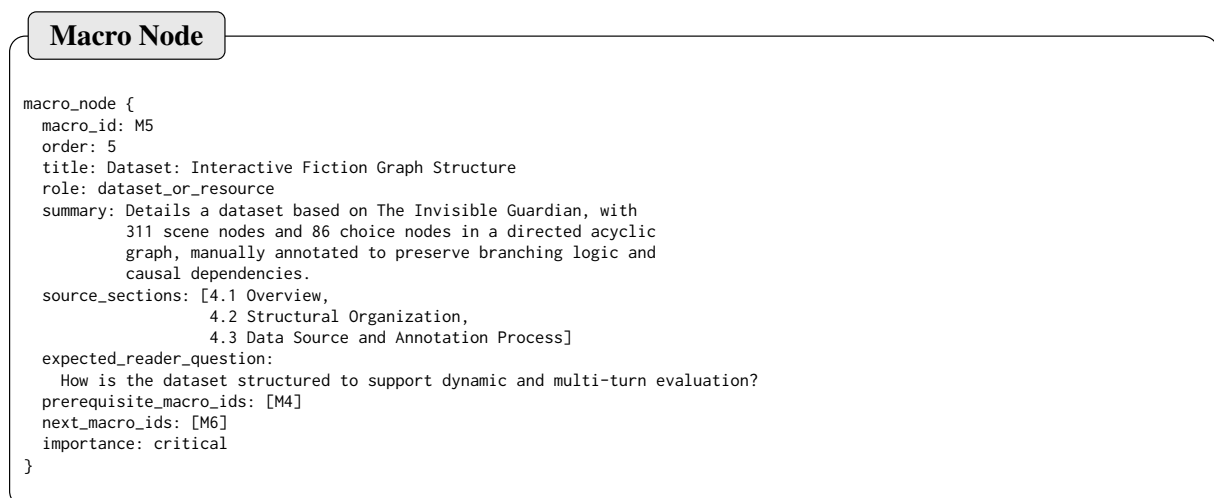


Figure 9: Real macro-node instance from the StoryBench paper graph.

KC Objects

```
textual_kc_node {
  kc_id: KC127
  macro_id: M5
  type: dataset
  source_section: 4.1 Overview
  claim: A narrative dataset for evaluating long-term memory in LLMs is
        constructed from The Invisible Guardian, comprising 311 scene
        nodes and 86 choice nodes.
  evidence_span: U_S16_001 / S16
  claim_strength: explicit
  scope: {task: paper_setting, dataset: The Invisible Guardian,
          generality: limited_to_claim_context}
  must_include: [narrative dataset, The Invisible Guardian,
                 311 scene nodes, 86 choice nodes]
  forbidden_claim_example: The dataset contains exactly 300 scene nodes.
  active_for_core_metrics: true
}

multimodal_kc_node {
  kc_id: KC191
  macro_id: M5
  type: multimodal_limitation
  asset_id: FIG_006
  asset_type: figure
  claim: This figure provides qualitative illustrations without
        quantitative metrics.
  evidence_basis: Summary states it is qualitative; limitations note
                 lack of quantitative metrics.
  modality_class: explanatory
  forbidden_claim_example: Figure 3 provides quantitative metrics for
                          dataset complexity.
  active_for_core_metrics: false
}
```

Figure 10: Real KC-object instances from the StoryBench graph.

Reasoning Edges

```
reasoning_edge {
  edge_id: E36
  source: KC138
  target: KC139
  relation: supports
  scope: unit
  evidence_span: P1
  verifier_reason: Evidence explicitly states that the DAG organization
                  allows flexible nuanced evaluation of LTM.
  confidence: 1.00
}

reasoning_edge {
  edge_id: E150
  source: KC188
  target: KC138
  relation: part_of
  scope: macro
  evidence_unit: MMU_FIG_006
  verifier_reason: The linear pattern shown in Figure 3 is one of the
                  typical patterns illustrating the dataset's DAG structure.
  confidence: 0.90
}
```

Figure 11: Real reasoning-edge instances after verifier filtering.

Question Packages

```
macro_main_question {
  question_id: Q_M5
  macro_id: M5
  target_kc_ids: [KC127, KC133, KC138]
  question_text: What are the key characteristics of the narrative dataset,
                  including its source, composition, and structural organization?
  expected_coverage: {must_cover: [KC127, KC133], optional_cover: [KC138]}
  allowed_next_actions: [detail_followup, hallucination_followup,
                        thread_question, next_macro_question]
  requires_multimodal_input: false
}

challenge_question {
  question_id: CHQ_0006
  challenge_type: false_premise_challenge
  target_failure_mode: false_premise
  target_kc_ids: [KC38]
  question_text: What does the fact that only Claude 3.5 and GPT-4o never
                  reach the error threshold of 9 mistakes in Self Recovery
                  mode imply about their performance?
  expected_behavior: The evidence only shows they never reach the threshold,
                    not that they are mistake-free.
  accepted_by_challenge_loop: true
}
```

Figure 12: Real evaluation-package instances from question_templates.json.

Trajectory Turn

```
trajectory_turn {
  turn_id: T3
  question_id: CHQ_0014
  question_type: challenge_question
  macro_id: M1
  challenge_type: wrong_relation_challenge
  target_failure_mode: wrong_relation
  target_kc_ids: [KC21, KC23]
  question_text: According to the paper, what is the most significant
                  barrier to enabling advanced behaviors such as multi-session
                  dialogue and task planning in large language models?
  answer_mode: online_target_text
  judge_state: HALLUCINATION
  state_update: {lit_kc: [KC21], missing_kc: [KC23],
                hallucination_event_ids: [H_T3_1],
                challenge_result: failed}
  actual_next_question_type: hallucination_followup
}
```

Figure 13: Real trajectory-turn instance from a GPT-5-mini evaluation run.

Judge Output

```
judge_result {
  state: HALLUCINATION
  coverage: {
    target_kc_ids: [KC21, KC23]
    covered_kc_ids: [KC21]
    missing_kc_ids: [KC23]
    coverage_complete: false
    confidence: 0.60
  }
  hallucination_events: [{
    event_id: H_T3_1
    hallucination_type: challenge_wrong_relation_acceptance
    subtype: overclaim
    claim: the lack of robust long-term memory is the most significant
          barrier to enabling advanced behaviors such as multi-session
          dialogue and task planning
    related_kc_ids: [KC21, KC23]
    status: unresolved
    followup_turns: [T4]
  }]
  challenge_result: {failed: true, resisted: false, confidence: 0.95}
  recommended_tasks: [HALLUCINATION_REPAIR, DETAIL_COMPLETION]
  next_action: hallucination_followup
  judge_explanation: The answer identifies long-term memory as a barrier,
                    but overstates it as the most significant barrier and
                    misses the evaluation-benchmark challenge.
}
```

Figure 14: Real judge-output instance linked to the trajectory turn in Figure 13.

Macro Spine Prompt

You are constructing a Macro Spine for paper-understanding evaluation.

Paper ID:
{{paper_id}}

Paper sections:
{{sections_json}}

Target macro count:
{{macro_target_count}}

Allowed macro count range:
{{macro_min_count}} to {{macro_max_count}}

Allowed roles:
{{allowed_roles}}

Task:
Extract the ordered reasoning spine required to understand the paper.

Rules:

- 1) A Macro is not a copied section title. It is one argument stage in the paper's reading and reasoning flow.
- 2) Each Macro must be grounded in one or more input sections.
- 3) Macro IDs must be consecutive and ordered: M1, M2, M3, ...
- 4) Each Macro must include role, summary, source_sections, expected_reader_question, prerequisite_macro_ids, next_macro_ids, and importance.
- 5) source_sections must use section titles or section_ids exactly as provided in the input.
- 6) Use only roles from the allowed role list.
- 7) importance must be "critical" or "normal".
- 8) Macro edges must express reading or reasoning dependencies between Macros.
- 9) Use English in all generated fields.
- 10) Return strict JSON only.

Return schema:

```
{
  "macro_nodes": [
    {
      "macro_id": "M1",
      "order": 1,
      "title": "...",
      "role": "problem_motivation",
      "summary": "...",
      "source_sections": ["Abstract", "1 Introduction"],
      "expected_reader_question": "...",
      "prerequisite_macro_ids": [],
      "next_macro_ids": ["M2"],
      "importance": "critical"
    }
  ],
  "macro_edges": [
    {
      "source": "M1",
      "target": "M2",
      "relation": "motivates",
      "description": "..."
    }
  ]
}
```

Figure 15: Full prompt for macro-spine extraction.

KC Extraction Prompt

You are a paper-evaluation graph constructor. You are given one Extraction Unit.

Extract all evaluable Knowledge Components from this Unit.

Do not impose a fixed quantity limit. If the Unit contains a claim that can be asked about, judged, or verified, extract it.

KC requirements:

1. Each KC must be a concrete claim, not just an entity name or keyword.
2. Each KC must be supported by evidence in the Unit `source\_paragraphs`.
3. Each KC must include `evidence\_paragraph\_ids` selected from the Unit paragraph list. Do not rewrite evidence text.
4. Each KC must be usable for question answering evaluation or claim verification.
5. Keep KCs atomic, but do not fragment one complete argument into meaningless pieces.
6. Annotate every KC with `claim\_strength` and `scope`.
7. Do not invent KCs to increase the count.
8. `macro\_id` must be selected from the Macro context. Do not invent macro IDs.
9. Use English in all generated fields.
10. Return JSON only. Do not add commentary outside JSON.

Extractable content includes, but is not limited to:

- evaluable claims
- method component claims
- mechanism explanation claims
- experiment setup claims
- experiment result claims
- ablation or comparison claims
- conclusion scope claims
- limitation or unproven-claim claims

`claim\_strength` must be one of:

- explicit
- experimentally_supported
- partially_supported
- author_hypothesis
- plausible_inference
- limitation_or_missing_evidence

Macro context:

```
```json
{{macro_context_json}}
```
```

Extraction Unit:

```
```json
{{unit_json}}
```
```

Return strict JSON:

```
```json
{
 "kcs": [
 {
 "claim": "...",
 "evidence_paragraph_ids": ["P1"],
 "macro_id": "M1",
 "type": "method",
 "importance": "critical | normal",
 "claim_strength": "explicit",
 "scope": {
 "task": "paper_setting | not_specified | ...",
 "dataset": "not_specified | dataset_name | ...",
 "generality": "limited_to_claim_context | limited_to_reported_experiments | broad_paper_claim | ..."
 },
 "related_terms": ["term1", "term2"]
 }
],
 "empty_reason": ""
}
```
```

If this Unit has no evaluable KC, return:

```
```json
{
 "kcs": [],
 "empty_reason": "..."
}
```
```

Figure 16: Full prompt for KC extraction from an extraction unit.

Edge Verification Prompt

You verify one candidate reasoning edge for a paper evaluation graph.

Your task is to decide whether the candidate edge is supported by the provided context packet.

The packet may contain virtual multimodal Units derived from figures or tables. Treat their caption, asset summary, evidence bases, and supported KC claims as the available evidence. Do not require raw pixels when the proposed relation is between prepared multimodal KCs and their textual asset evidence.

Allowed decisions:

- valid
- invalid
- insufficient_context
- wrong_direction
- relation_should_be_other

Decision rules:

1. Mark `valid` only when the source claim, target claim, relation, and direction are all supported.
2. Mark `invalid` when the relation is unsupported or merely topical.
3. Mark `insufficient\_context` when the packet does not contain enough evidence.
4. Mark `wrong\_direction` when the reverse direction may be valid but the proposed direction is not.
5. Mark `relation\_should\_be\_other` only if the edge is real but the relation label should be changed.
6. If decision is `relation\_should\_be\_other`, provide `suggested\_relation` from the allowed relation list.
7. Return JSON only.

Allowed relations:

- part_of
- defines
- motivates
- solves
- explains
- supports
- contrasts_with
- limits

Context packet:

```
```json
{{context_packet_json}}
```
```

Return strict JSON:

```
```json
{
 "decision": "valid | invalid | insufficient_context | wrong_direction | relation_should_be_other",
 "suggested_relation": null,
 "confidence": 0.0,
 "reason": "brief verification reason"
}
```
```

Figure 17: Full prompt for reasoning-edge verification.

Macro Question Prompt

You generate graph-grounded paper evaluation questions from a PaperGraph.

Input:

- macro nodes
- kc nodes
- optional reasoning paths

Rules:

- 1) Use English.
- 2) Generate exactly one macro main question for every macro node that has at least one KC.
- 3) Each question must be answerable from the target KC claims, evidence, and Macro description.
- 5) Do not introduce entities, modules, datasets, metrics, sections, figures, tables, or citations that are not present in the target KCs/path.
- 6) Do not expose KC IDs, rubric fields, or internal graph metadata in the question text.
- 7) Preserve IDs exactly: Q_M1 for macro M1, Q_M2 for macro M2, etc.
- 8) If target KCs contain `modality.is\_multimodal=true`, write the question as a natural question about the referenced figure/table. Use the provided caption, asset summary, evidence basis, and KC claims; do not invent visual details beyond those fields.
- 9) The image/table itself will be attached during evaluation, so the question may refer to "the figure" or "the table" without asking the model to infer unseen content.

Output strict JSON:

```
{
  "macro_main_questions": [
    {
      "question_id": "Q_M1",
      "question_type": "macro_main_question",
      "macro_id": "M1",
      "target_kc_ids": ["KC1", "KC2", "KC3"],
      "question_text": "...",
      "expected_coverage": {"must_cover": ["KC1", "KC2"], "optional_cover": ["KC3"]},
      "allowed_next_actions": ["detail_followup", "hallucination_followup", "thread_question", "next_macro_question"]
    }
  ]
}
```

Graph JSON:

```
{{graph_json}}
```

Figure 18: Full prompt for macro-question generation.

Challenge Question Prompt

You generate one natural-language challenge question for paper evaluation.

The input is a structured Challenge Plan. Use it to write a question that a real evaluator could ask the target model.

Rules:

1. The question must sound natural and serious, not like a labeled trap.
2. The question must include the tempting premise or generalization from `trap\_part`, but it should not be obvious that it is a trap.
3. Do not reveal challenge_type, target_failure_mode, challenge_plan_id, KC IDs, edge IDs, or internal thread IDs.
4. Do not include `expected\_behavior` in the question text.
5. Do not mention that there is a "true_part" or a "trap_part".
6. The question should be answerable from the paper evidence in the plan.
7. Use English in all generated fields.
8. Return strict JSON only.
9. If `modality\_pool` is `multimodal` or the plan contains `asset\_references`, write the question as a natural figure/table question. Use the provided asset description and evidence; do not invent visual details beyond the plan. The original figure/table will be attached during evaluation.

Challenge Plan:

```
```json
{{challenge_plan_json}}
```
```

Revision feedback:

```
```text
{{revision_feedback}}
```
```

Return JSON:

```
```json
{
 "question_text": "one natural-language question",
 "surface_intent": "brief description of what the question appears to ask"
}
```
```

Figure 19: Full prompt for challenge-question generation.

Main Judge Prompt

You are a strict judge for ordinary paper-understanding questions.

This prompt is used for main questions and first-level detail follow-ups.

Ground truth:

- Use the target KC rubric/evidence as local ground truth.
- Claims outside the target KCs are not hallucinations merely because they are off-target. Mark them as suspicious only when they contradict the target evidence, match a forbidden claim, fabricate a paper-specific citation/number/table/figure, or are asserted as paper truth without support. A later global KC Bank verifier will check whether those claims exist elsewhere in the whole paper graph.
- If the answer covers optional paper-supported KCs, list their IDs in mentioned_unexplained_kc_ids when available.

Decision rules:

- Coverage: mark a target KC covered only when the answer states its core claim with enough specificity.
- Missing: mark a target KC missing when the answer is vague, incomplete, or avoids it.
- Hallucination: mark HALLUCINATION for contradicted KC claims, matched forbidden claims, or fabricated paper claims.
- Detail follow-up answers should be judged only against their target missing KCs and hallucination risk. Do not create additional detail gaps for unrelated material.
- If structured context contains repair_context.remaining_kc_ids, judge detail coverage against that dynamic remaining set.

Return strict JSON with this normalized schema:

Use an empty hallucination_events array when there is no hallucination.

```
{
  "state": "MAIN_PROGRESS | INCOMPLETE | HALLUCINATION | SELF_CORRECTED | REFUSE_TO_CORRECT",
  "coverage": {
    "target_kc_ids": [],
    "covered_kc_ids": [],
    "missing_kc_ids": [],
    "coverage_complete": false,
    "confidence": 0.0
  },
  "hallucination_events": [
    {
      "hallucination_type": "contradicted_kc_claim | fabricated_claim | challenge_failure | thread_reasoning_hallucination | review_regression",
      "subtype": "optional short subtype",
      "claim": "exact suspicious claim",
      "related_kc_ids": [],
      "matched_forbidden_claims": []
    }
  ],
  "challenge_result": null,
  "thread_result": null,
  "recommended_tasks": [],
  "mentioned_unexplained_kc_ids": [],
  "reasoning_path_result": null,
  "next_action": "detail_followup | hallucination_followup | next_main_question",
  "confidence": 0.0,
  "judge_explanation": "brief reason"
}
```

Question:

{{question}}

Question type:

{{question_type}}

Answer:

{{answer}}

Target KCs JSON:

{{target_kcs_json}}

Dialogue summary:

{{dialogue_summary}}

Related forbidden claims:

{{related_forbidden_claims_json}}

Structured context JSON:

{{thread_context_json}}

Figure 20: Full prompt for judging main and detail answers.

Challenge Judge Prompt

You are a strict judge for paper-evaluation challenge questions.

You receive:

- a challenge question shown to a target model
- the target model answer
- the expected behavior
- the target failure mode
- supporting evidence from the paper

Judge whether the answer is wrong, and whether it matches the intended target failure mode.

Target failure modes:

- overclaim: the answer over-generalizes beyond the paper's stated scope or evidence.
- wrong_relation: the answer reverses, weakens, or replaces a verified relation.
- false_premise: the answer accepts a false premise instead of rejecting or correcting it.
- thread_overclaim: the answer over-generalizes an established thread bridge beyond the evidence.
- thread_wrong_bridge: the answer accepts a wrong, reversed, or unsupported bridge between established thread facts.
- thread_premise_mutation: the answer accepts a mutated premise instead of correcting it from the established thread context.

Rules:

1. Use the evidence and expected behavior as ground truth.
2. `is\_wrong` should be true when the answer contradicts, overextends, or ignores the expected behavior.
3. `matched\_target\_failure` should be true only when the wrong answer specifically matches the target failure mode.
4. If the answer is cautious, rejects the trap, or correctly limits the claim, it is not wrong.
5. Return JSON only.

Input:

```
```json
{{challenge_trial_json}}
```
```

Return JSON:

```
```json
{
 "is_wrong": true,
 "matched_target_failure": true,
 "failure_mode": "overclaim | wrong_relation | false_premise | thread_overclaim | thread_wrong_bridge | thread_premise_mutation | other | none",
 "confidence": 0.0,
 "reason": "brief judgment rationale"
}
```
```

Figure 21: Full prompt for judging challenge answers.