

RPC-BENCH: A FINE-GRAINED BENCHMARK FOR RESEARCH PAPER COMPREHENSION

Anonymous authors

Paper under double-blind review

ABSTRACT

Leveraging large foundation models for document understanding has emerged as a rapidly advancing research area. Unlike general-purpose documents, research papers constitute a particularly challenging domain, as they are characterized by complex figures, detailed tables, and highly specialized scientific knowledge. However, existing benchmarks pay limited attention to evaluating the fine-grained capabilities of current models in comprehending research papers at scale. To address this gap, we propose RPC-Bench, a large-scale fine-grained question-answering benchmark constructed from review-rebuttal exchanges of high-quality academic papers, with each paper available in two input formats (pure text and rendered page images) enabling evaluation of both large language models (LLMs) and visual language models (VLMs). We design a fine-grained taxonomy aligned with the research flow of academic papers to guide annotation. We also define an elaborate LLM-human interaction annotation framework to support large-scale labeling and quality control. Following the LLM-as-a-Judge paradigm, we develop a scalable framework that evaluates models on correctness-completeness and conciseness, with high agreement to human judgment. Experiments show GPT-5 leads with a 66.54% correctness-completeness score, dropping to 35.05% after conciseness adjustment. In addition, multimodal LLMs perform better on pure text than visual-text inputs, highlighting the need for improved visual integration in scholarly document understanding. Our code and data is available¹.

1 INTRODUCTION

Large foundation models are increasingly serving as research copilots, supporting tasks such as deep research (Schmidgall et al., 2025; Cappello et al., 2025), paper reviewing (Chang et al., 2025; Zhang et al., 2025), and even the automation of the research process itself. A critical prerequisite for these applications is the ability of both large language models (LLMs) and vision-language models (VLMs) to achieve a comprehensive understanding of research papers. Beyond parsing explicit content, such models must be able to grasp specialized concepts, analyze methodological motivations, and evaluate experimental limitations that can guide subsequent scientific discovery.

Although document understanding has made remarkable progress in recent years, existing works have largely focused on general-domain tasks such as layout analysis, content localization, and caption generation. In contrast, research paper understanding is substantially more complex: it demands domain-specific expertise to interpret technical concept, specialized methodologies, and detailed experimental designs. For example, PeerQA (Baumgärtner et al., 2025) is restricted in scale, covering only a small number of question-answer (QA) pairs. SPIQA (Pramanick et al., 2024), DocGenome (Xia et al., 2024), and ArXivQA (Li et al., 2024b) rely heavily on synthetic QA rather than real scholarly interactions, limiting their authenticity. More broadly, these benchmarks are constrained by task-based taxonomies without categorizing questions by the depth of content understanding. They also employ limited evaluation metrics and often lack simultaneous support for both textual and visual inputs. As shown in Table 1, there is no comprehensive benchmark that evaluates the ability of various methods to deeply understand large-scale research papers.

To address this gap, we introduce **RPC-Bench**, a large-scale benchmark designed for in-depth research paper comprehension. RPC-Bench is constructed from high-quality publications

¹<https://anonymous.4open.science/r/PRC-Bench-B327>

Table 1: Comparison with the most relevant research paper Benchmarks. Conc.=Conciseness; Corr.=Correctness; F1-like is computed as the harmonic mean of correctness and completeness; inp.=input. “Eval. Metrics” are LLM-based metrics.

Benchmarks	Papers	QA	Real QA	Taxonomy	Eval. Metrics	Textual inp.	Visual inp.
PeerQA	208	579	✓	task	Corr.	✓	✗
SPIQA	25.5K	270K	✗	task	LLMLogScore	✓	✓
ArXivQA	16.6K	-	✗	task	-	✗	✓
DocGenome	500K	-	✗	task	GPT-acc	✗	✓
RPC-Bench	4050	46.3K	✓	content	Conc., F1-like	✓	✓

(2013–2024) on OpenReview², together with their associated review–rebuttal exchanges from the peer-review process. Unlike synthetic datasets, our QA pairs are directly derived from these authentic review–rebuttal interactions and transformed into question–answer format through a collaborative LLM–human workflow, ensuring both diversity and reliability. After rigorous filtering, the final benchmark encompasses 4,050 papers and 46.3K QA pairs.

To systematically capture the essential aspects of paper understanding, we decompose the research workflow into a fine-grained taxonomy of three primary dimensions (Concepts, Methods, and Experiments) further divided into nine categories. This taxonomy provides principled guidance for annotation and evaluation, enabling nuanced assessment across the full spectrum of academic paper comprehension.

In addition, we design a scalable evaluation framework based on LLM judges that align with human judgment, supporting both pure-text and rendered-page inputs to benchmark LLMs and VLMs alike. Model outputs are evaluated jointly on correctness (accuracy of generated responses, akin to precision), completeness (coverage of essential content, akin to recall), and conciseness, with multiple pilot-tested LLM judges aggregated to produce stable, human-consistent scores.

We conduct extensive experiments across 19 state-of-the-art models, including 6 LLMs, 3 document-centric models (DCMs), 5 VLMs, and 5 retrieval-augmented generation (RAG) models. Our results reveal that none of the evaluated models are capable of fully comprehending research papers. Even the best model, GPT-5, achieved only 66.54% on F1-like (harmonic mean of correctness and completeness), which dropped to 35.05% under the conciseness-constrained F1-like. Furthermore, for multimodal-capable LLMs, replacing text inputs with page-image inputs consistently reduced F1-like by 4.57–33.40%, highlighting the current weakness of models in visual reasoning over scholarly documents. In summary, our contributions are:

- We introduce RPC-Bench, the first large-scale benchmark grounded in authentic review–rebuttal exchanges, featuring a fine-grained taxonomy aligned with the research workflow to enable systematic annotation and evaluation of research paper comprehension.
- We introduce a LLM–human collaborative annotation framework that supports large-scale QA generation and rigorous quality control.
- We develop a scalable evaluation framework that jointly measures the balance among correctness, completeness, and conciseness with high consistency to human judgment.
- We conduct a comprehensive empirical study on 19 advanced models, identifying fundamental limitations in both text-based and multimodal research paper understanding.

2 RELATED WORK

Methodologies for Document Question Answering. The current landscape of document QA is organized around three complementary pillars: (i) large foundation models, (ii) document-centric architectures, and (iii) retrieval-augmented generation (RAG). State-of-the-art LLMs and VLMs include proprietary models like GPT-5 (Leon, 2025), Claude 4.1, and Gemini 2.5 (Comanici et al., 2025), as well as leading open-source models such as the Qwen (Yang et al., 2025), GLM (GLM et al., 2024), and DeepSeek (Deng et al., 2025) series.

Document-centric architectures are introduced to accommodate to the structural and layout peculiarities of long documents. One line of the work, such as Monkey-Chat-7B and DocOwl2-8B (Li

²<https://openreview.net/>

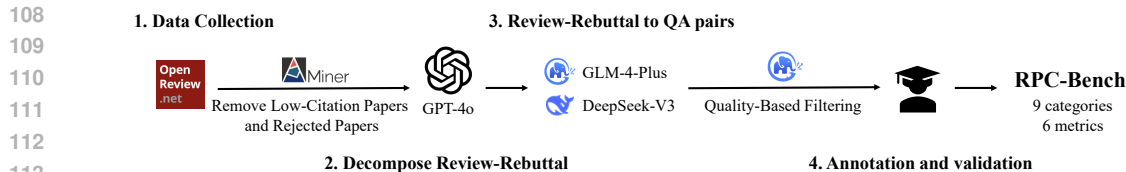


Figure 1: RPC-Bench Construction Pipeline. Papers and review–rebuttal pairs are crawled from OpenReview, low-citation and rejected papers are removed via AMiner, segmented into comment–response units with GPT-4o, rewritten into QA pairs using GLM-4-Plus and DeepSeek-V3, and low-quality QA items are discarded before iterative human annotation and review.

et al., 2024c; Hu et al., 2024), enables direct, OCR-free document understanding, avoiding error propagation from external OCR. Another line, exemplified by layout-aware models like **DocLLM** Wang et al. (2023) and **Docopilot-8B** (Duan et al., 2025), explicitly models two-dimensional page layout to better parse complex structures like tables and forms.

RAG-based approaches mitigate the limited parametric knowledge of models when operating over vast corpora by grounding generation in retrieved evidence. Textual RAG methods include **RAPTOR** (Sarathi et al., 2024), which uses recursive clustering, as well as **HippoRAG** (Gutiérrez et al., 2025) and **MemoRAG** (Qian et al., 2025), which optimize indexing and memory. More broadly, multimodal RAG variants such as **VisRAG** (Yu et al., 2024) and **VDocRAG** (Tanaka et al., 2025) extend retrieval to visual content, enabling evidence discovery within figures and tables. The RAG ecosystem is further supported by toolkits like **FlashRAG** (Jin et al., 2025) and explorations into alternative data structures such as knowledge graphs with **GraphRAG** (Edge et al., 2025).

Document QA Benchmarks. Numerous benchmarks have been developed to standardize evaluation. For instance, SPIQA (Pramanick et al., 2024) focuses on multimodal questions concerning figures and tables in scientific papers. DocGenome (Xia et al., 2024) provides a large-scale, multi-domain dataset for both pre-training and high-level evaluation. LongDocURL (Deng et al., 2024) marks a step toward finer granularity by assessing the distinct skills of understanding, reasoning, and locating. PeerQA (Baumgärtner et al., 2025), akin to our work, is limited to textual data with a relatively small set of annotated QA pairs, thereby neglecting the crucial multimodal elements inherent in research papers. However, existing benchmarks suffer from several notable limitations. (1) **Limited scale:** Existing benchmarks are often small in scope, failing to cover large collections of research papers and question–answer pairs. (2) **Quality issues:** Many rely on automatically generated QA pairs whose correctness is hard to guarantee, and they usually classify tasks only by type rather than by the depth or nature of question content. (3) **Narrow and shallow evaluation:** Current benchmarks tend to emphasize multimodal QA while lacking systematic assessment of paper understanding and reasoning. Moreover, they typically focus on single-dimensional metrics such as accuracy, which cannot fully capture the quality of long-text answers.

Unlike prior works, our benchmark offers large-scale, realistic academic QA grounded in peer reviews and rebuttals. We categorize questions according to the stages of academic research and assess a broad spectrum of document QA methods. Furthermore, we transform eligible peer-review QA pairs into binary classification tasks. Our comprehensive evaluation reveals that current approaches still exhibit pronounced limitations in expert-level comprehension of scholarly literature.

3 RPC-BENCH

RPC-Bench is a challenging benchmark for research paper comprehension in realistic settings, designed to evaluate the ability to accurately comprehend complex domain-specific knowledge and to reason over scholarly content. (1) curating large-scale open-ended QA pairs and yes/no QA pairs from high-quality papers and their review–rebuttal exchanges; (2) introducing a fine-grained taxonomy and rigorous metrics for systematic assessment; and (3) defining a carefully designed LLM–human interactive annotation framework to enable large-scale labeling and strict quality control. The following sections detail the data collection, taxonomy design, annotation process, and evaluation protocol. Figure 1 presents the overall framework for benchmark construction.

3.1 DATA COLLECTION

To rigorously assess model capabilities in paper understanding, we built a three-stage data pipeline:

- **Broad coverage collection:** Collected 44.7K publicly available peer-reviewed papers and their corresponding review–rebuttal pairs from OpenReview³ spanning 2013–2024.
- **Quality refinement:** Matched the collected papers against the popular academic search and mining system AMiner⁴ to remove incomplete entries, resulting in a curated set of 17.7K papers.
- **Impact-based sampling:** Selected 3521 accepted papers with ≥ 50 citations as positive samples, plus 361 highly-cited and 361 random rejected papers as challenging negatives to enhance the robustness and generalization of the benchmark.

This pipeline yields a scholarly collection of 4243 papers. We chronologically split this collection as follows: 3556 papers for training, 487 papers for validation, and 200 papers for testing.

3.2 TAXONOMY DESIGN

Our goal is to assess whether models truly comprehend and reason about scholarly articles, rather than relying on memorization or surface pattern-matching. To this end, we designed a taxonomy guided by the natural research flow of academic papers. It begins with *what-questions*, which focus on clarifying fundamental concepts, terminology, and contextual background. It then advances to *how-questions*, which explore the mechanics of methods, algorithmic details, and experimental setups. Finally, it deepens into *why-questions*, which examine the underlying motivations of methods and the reasoning behind observed experimental outcomes. By structuring questions in this layered way—moving from foundational understanding, to technical mechanics, to deeper rationale—the taxonomy not only reveals the strengths of existing works but also highlights their limitations. These insights, in turn, help identify open gaps and inspire new directions for future research.

Task Taxonomy

1. **Concept Understanding (C.U.) [What]:** Clarifies or explains key concepts, terminology, theoretical viewpoints, or information conveyed in figures, tables, or formulas.
2. **Methods**
 - 2.1. **Method Disambiguation (M.D.) [What]:** Clarifies methodological details to resolve misunderstandings or ambiguities, ensuring an accurate grasp of proposed approaches.
 - 2.2. **Method Mechanics (M.M.) [How]:** Questions about the implementation or function of methodological workflow or components, such as the effect of specific modules in models.
 - 2.3. **Motivation Analysis (M.A.) [Why]:** Examines the rationale, principles, or intentions underlying a proposed method or decision.
 - 2.4. **Method Comparison (M.C.):** Compares the proposed approach with baseline methods, analyzing similarities, differences, or performance to highlight novelty.
3. **Experiments**
 - 3.1. **Experimental Exposition (E.E.) [What]:** Describes experimental outcomes, infers how modifications or variations could impact results or conclusions, and addresses reasoning tasks such as calculation, counting, or comparative analysis.
 - 3.2. **Experimental Setup (E.S.) [How]:** About the design, configuration, and execution of experiments.
 - 3.3. **Experimental Analysis (E.A.) [Why]:** Studies the reasons of specific experimental outcomes, links them to the proposed approach, and assesses their generalizability and potential impact.
4. **Claim Verification (C.V.):** Binary classification tasks that assess the correctness of claims, hypotheses, or experimental conclusions.

Based on this principle, we define a four-granularity taxonomy organized around key components of research papers (see taxonomy above), enabling fine-grained and multi-perspective coverage of academic paper understanding. Most categories are formulated as free-form QA tasks, while the

³<https://openreview.net/>

⁴<https://www.aminer.cn/>

216 Verification category is implemented as a binary classification task. Both formats require models to
 217 locate, integrate, understand, and reason over information drawn from the target paper.
 218

219 3.3 ANNOTATION PROCESS

220
 221 Manual annotation of taxonomy-based QA pairs requires domain expertise and significant time for
 222 labeling and verification, making large-scale, high-quality data collection prohibitively costly. To
 223 mitigate this, we propose a semi-automated pipeline that leverages multiple large language models
 224 (LLMs) in a hybrid manner, reducing human effort while maintaining annotation quality.

225 Crawled review–rebuttal pairs from OpenReview usually contain overall reviews and general replies
 226 rather than paired comment–response matches. To enable fine-grained question generation, we first
 227 use GPT-4o (chosen for its strong reasoning and contextual understanding) to decompose each re-
 228 view into minimal, self-contained comment–response pairs. Guided by our taxonomy, GLM-4-Plus
 229 and DeepSeek-V3 then rewrite these pairs into free-form QA or claim verification tasks, assigning
 230 each to the proper taxonomy category. This setup, validated in a pilot study, achieves competitive
 231 rewriting quality at far lower cost than GPT-4o, making it practical for large-scale generation.

232 Finally, to ensure the reliability of the automatically generated questions, we apply a two-stage
 233 filtering process using GLM-4-Plus: review-rebuttal filtering and QA-level filtering. The filtering
 234 criteria exclude entries lacking substantive academic content, including purely mechanical changes,
 235 reliance on external resources, indirect or unanswered replies, and superficial commitments without
 236 concrete improvements. Detailed criteria and examples are provided in Appendix C.1.

237 We employed four annotators (Master’s degree or above), with two handling annotation and two
 238 reviewing. Before formal annotation, all annotators received training and practiced QA conversion
 239 on 10 sample papers, with iterative feedback until each reached a $\geq 95\%$ pass rate. To ensure quality
 240 over speed, annotators were limited to 80 QA pairs per day, averaging 5–6 minutes per question.
 241 Annotated data were promptly reviewed, and problematic cases were returned for correction.

242 Using the annotation platform (Appendix
 243 C.2), annotators examined each segmented re-
 244 view–rebuttal pair and chose the better output
 245 between GLM-4-Plus and DeepSeek-V3, while
 246 verifying taxonomy labels. If both outputs were
 247 inadequate, they rewrote the pair manually and
 248 assigned the correct category. To reduce bias,
 249 model identities were anonymized as Model1
 250 and Model2, with randomized ordering. Anno-
 251 tators could discard low-quality pairs or gener-
 252 ate multiple sub-questions from a single pair,
 253 provided each addressed a distinct aspect. The
 254 review platform (Appendix C.3) displayed both
 255 original and rewritten content, allowing review-

256 ers to approve or reject entries. Rejections re-
 257 quired specific feedback to guide further revisions. Due to cost constraints, only the validation and
 258 test sets were manually annotated, while the training set retained QA pairs generated by LLMs. Ta-
 259 ble 2 summarizes the dataset statistics, and Figure 2 presents the distribution of data across different
 260 categories and domain. The full set of prompts used in this stage, along with the graphical user
 261 interface for annotation and review, are provided in Appendix C.

262 3.4 QUALITY CONTROL

263 Using a two-stage semi-automated pipeline combining LLM-based filtering and expert validation,
 264 we ensured annotation quality and retained only questions answerable from the paper’s content, thus
 265 ensuring the RPC-Bench reliably assesses research paper comprehension.
 266

267 As described in Section 3.3, GLM-4-Plus was employed to remove low-quality or unanswerable
 268 data from both review–rebuttal pairs and converted QA items across the entire dataset, including
 269 the training set. This step eliminated 24.5% of review–rebuttal pairs and 12.96% of QA items,
 respectively.

Table 2: Statistics of the RPC-Bench. A/M Q: average/max question length. A/M A: average/max answer length. Lengths are measured in words.

Statistics	train	val	test
Papers	3427	423	200
Accept	2218	314	116
year	2013-2022	2022-2023	2024
Venue	15	7	4
QA	39203	6152	2787
A/M Q	25.4/157	24.9/226	24.2/250
A/M A	72.4/320	93.9/1337	87.9/773

270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323

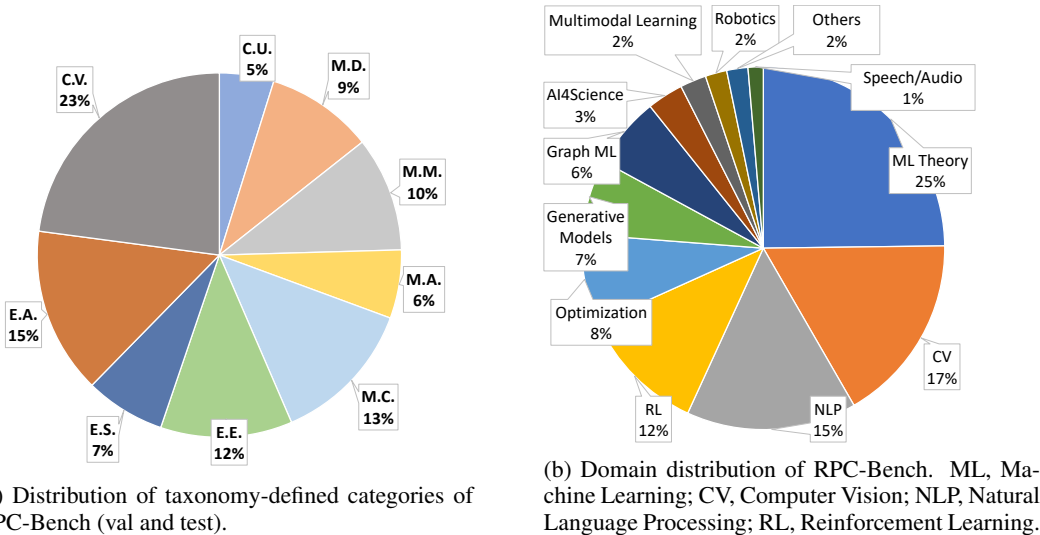


Figure 2: Comparative overview of category and domain compositions in RPC-Bench.

During annotation, reviewers inspected the annotated data, returned problematic cases to the original annotators for correction, and jointly verified question answerability. This stage further removed 8.87% of QA items identified as low-quality or unanswerable based solely on the information in the corresponding paper. For QA items referencing specific numbers, formulas, or section indices, annotators and reviewers were required to verify their presence in the final paper version and update all indices accordingly to maintain positional accuracy and consistency.

3.5 EVALUATION PROTOCOL

To enable consistent and flexible assessment of academic paper understanding, we propose a standardized evaluation framework. For binary classification with clear ground-truth labels, accuracy (ACC) is used as the primary metric. For open-ended QA, traditional automatic metrics (e.g., BLEU, BERTScore) often fail to capture answer quality due to diverse valid semantically-equivalent expressions. Following recent work on LLM-as-a-Judge (D’Souza et al., 2025; Desmond et al., 2025; Li et al., 2024a), we adopt an LLM-based evaluation that scores responses along three dimensions: conciseness (brevity and without irrelevant detail), correctness (accuracy and fidelity, akin to precision), and completeness (coverage of essential content, akin to recall). Each is rated on a 0–5 scale. We further compute two derived metrics: an F1-like score (harmonic mean of correctness and completeness) and informativeness, the aggregate of all three dimensions.

$$\text{F1-like} = \frac{(1 + \beta^2) \times (\text{Correctness} \times \text{Completeness})}{\beta^2 \times \text{Correctness} + \text{Completeness}}, \quad \text{Informativeness} = \text{F1-like} \times \frac{\text{Conciseness}}{5}$$

where β controls the weight between correctness and completeness ($\beta = 1$ by default), while conciseness is normalized to [0,1]. This captures the F1-like balance of correctness and completeness, with conciseness penalizing verbosity. Full evaluation prompts are given in Appendix C.7.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate models on the RPC-Bench test set under two input settings: pure-text and image-based. For text, each PDF is converted to Markdown via MinerU⁵, with content truncated if it exceeds the model’s context window. For images, PDFs are rendered with PyMuPDF at 200 DPI, and the first 15 pages are used to balance coverage and context limits. For models without multi-image support (e.g., Monkey), these pages are concatenated into a single composite image for compatibility.

⁵<https://github.com/opendatalab/MinerU>

We evaluate 19 models across the two configurations. The set includes: **LLMs**: DeepSeek-V3.1 (Deng et al., 2025), GLM-4.5 (GLM et al., 2024), Qwen3 (qwen3-235b-a22b) (Bai et al., 2023), GPT-5 (gpt-5-2025-08-07) (Leon, 2025), Claude-4 (claude-sonnet-4-20250514) (Anthropic, 2025), Gemini-2.5 (gemini-2.5-pro) (Comanici et al., 2025); **Document-Centric Models (DCM)**: DocOwl2(V) (Hu et al., 2024), Docopilot(V) (Duan et al., 2025), Monkey(V) (Li et al., 2024c); **VLMs**: GLM-4.5V (GLM et al., 2024), Qwen3(V), GPT-5(V), Claude-4(V), Gemini-2.5-Pro(V); **RAG Models**: HippoRAG2 (Gutiérrez et al., 2025), MemoRAG (Qian et al., 2025), Raptor (Sarathi et al., 2024), VdocRAG(V) (Tanaka et al., 2025), VisRAG(V) (Yu et al., 2024). Here, “(V)” denotes image-based input.

To maximize performance while leveraging each model’s strengths, we impose minimal inference constraints: (1) answers must rely only on the given paper; (2) open-ended responses must be professional, concise, and under 3,000 characters; (3) claim verification outputs must be strictly True or False. Full prompts for two task types are provided in Appendix C.6.

As detailed in Section 3.5, open-ended QA is evaluated with ROUGE-L, BERTScore-F1, Conciseness, Correctness, Completeness, F1-like, and Informativeness, while Claim Verification is measured by accuracy. All results are reported on a standardized 0–100 scale.

4.2 MAIN RESULTS

Table 3 reports all model results on the RPC-Bench test set, highlighting the following findings:

Traditional surface-matching metrics are insufficient for evaluating paper comprehension, as they fail to capture true semantic understanding. For example, ROUGE-L and BERTScore cannot reliably distinguish large- from small-scale models (LLMs/VLMs vs. DCM/RAG < 10B). Some models achieve high lexical scores but perform poorly on semantic measures: Monkey(V) attains the best ROUGE-L (20.16%) and strong BERTScore (55.19%), yet its correctness and completeness fall to 17.43% and 11.54%. This shows surface metrics overestimate comprehension, while our framework evaluates semantic correctness and conciseness comprehensively.

Empirically, LLMs comprehend academic papers better with text-only than with image inputs. Despite multimodal support, the domain knowledge and intricate structures and layouts in text and images hinder accurate understanding. For example, Qwen3’s F1-like score falls from 53.68% (text-only) to 20.28% (image), with conciseness also dropping sharply. GPT-5(V) shows higher conciseness than its text-only version (61.85% vs. 52.68%), but this stems from shorter, less informative outputs caused by reduced correctness and completeness. Overall, the steepest declines appear in correctness and completeness, revealing that current multimodal models still struggle to exploit scholarly visual and textual information.

Academic paper comprehension is especially challenging for small-scale models (~8B). With limited capacity, document-centric models struggle to integrate full-paper content, yielding low F1-like scores (8–18%) and sometimes incoherent outputs—showing that fine-tuning on general-domain data is insufficient for our task. RAG models perform better via targeted retrieval, yet their correctness and completeness remain below 30, indicating limited grasp of essential information.

Current models struggle to balance correctness, completeness, and conciseness in paper-based QA. We use Informativeness, a composite metric capturing these dimensions, to assess overall ability. Even top closed-source models score modestly—GPT achieves only 35.05%. Open-source GLM-4.5 shows strong correctness and completeness (57.44%) but low conciseness (37.46%), reducing its Informativeness to 22.41%. These findings highlight significant room for improvement in academic paper comprehension.

4.3 PERFORMANCE ACROSS TASK CATEGORIES

Performance of Taxonomy-Defined Question Types. We conduct a fine-grained evaluation of paper comprehension across taxonomy-defined question types. As shown in Figure 3 (a), models perform better on simpler tasks (e.g., concept understanding, method discrimination) than on deeper reasoning tasks, with the gap widening for image-based inputs. Although figures and tables encode rich information, current models struggle to integrate them with long contexts into coherent reasoning. Overall, most models cannot effectively perform contextual reasoning across text and images

Table 3: Evaluation results of free-form QA on the test set. RG-L=ROUGE-L; B-S=BERTScore-F1; Compl. = Completeness; Info. = Informativeness. The best results are highlighted in **bold**, and the second-best results are indicated with underlining.

Model Type	Model	Traditional		LLM-as-judge				
		R-L	B-S	Conc.	Corr.	Compl.	F1-like	Info.
LLM	DeepSeek-V3.1	19.12	55.98	53.15	56.10	52.17	54.06	28.73
	GLM-4.5	16.03	53.18	39.02	57.55	57.32	57.44	22.41
	Qwen3	16.16	54.25	37.46	53.70	53.65	53.68	20.11
	GPT-5	16.89	54.52	52.68	67.66	65.44	66.54	35.05
	Claude-4	16.60	54.02	35.78	55.74	54.98	55.36	19.81
	Gemini-2.5	18.24	<u>55.67</u>	53.14	<u>60.54</u>	56.15	<u>58.26</u>	30.96
DCM	DocOwl2(V)	14.32	46.42	50.57	11.97	6.48	8.41	4.25
	Docopilot(V)	16.92	53.82	38.18	18.22	17.24	17.71	6.76
	Monkey(V)	20.16	55.19	56.70	17.43	11.54	13.88	7.87
VLM	GLM-4.5V	<u>19.66</u>	55.48	59.55	47.31	41.09	43.98	26.19
	Qwen3(V)	14.70	53.72	20.62	20.17	20.39	20.28	4.18
	GPT-5(V)	17.32	54.85	<u>61.85</u>	57.80	53.77	55.71	<u>34.46</u>
	Claude-4(V)	13.33	50.63	24.23	51.55	50.05	50.79	12.31
	Gemini-2.5(V)	17.27	54.85	50.53	47.22	43.24	45.14	22.81
RAG	HippoRAG2	18.71	54.16	42.37	31.83	26.57	28.97	12.27
	MemoRAG	13.55	52.70	49.81	23.60	18.21	20.56	10.24
	Raptor	18.35	54.00	35.16	23.92	18.83	21.07	7.41
	VdocRAG(V)	17.77	52.22	65.61	21.32	13.42	16.47	10.81
	VisRAG(V)	16.80	54.93	36.21	24.51	21.82	23.09	8.36

to address experimental specific questions (especially experimental analysis), underscoring the need for advances in multimodal paper understanding.

Results of Claim Verification QA Instances. Figure 3 (b) shows the accuracy of baselines on claim verification tasks. Overall, the baseline methods show limited accuracy in claim verification. We find that some multimodal models perform relatively well, likely because they can capture the overall meaning and key claims of a paper. However, large language models may struggle to identify crucial evidence from long contexts. Notably, certain models (e.g., Claude-4, HippoRAG2) exhibit weak instruction-following ability, often failing to output strictly “yes” or “no,” which undermines their fact-checking accuracy.

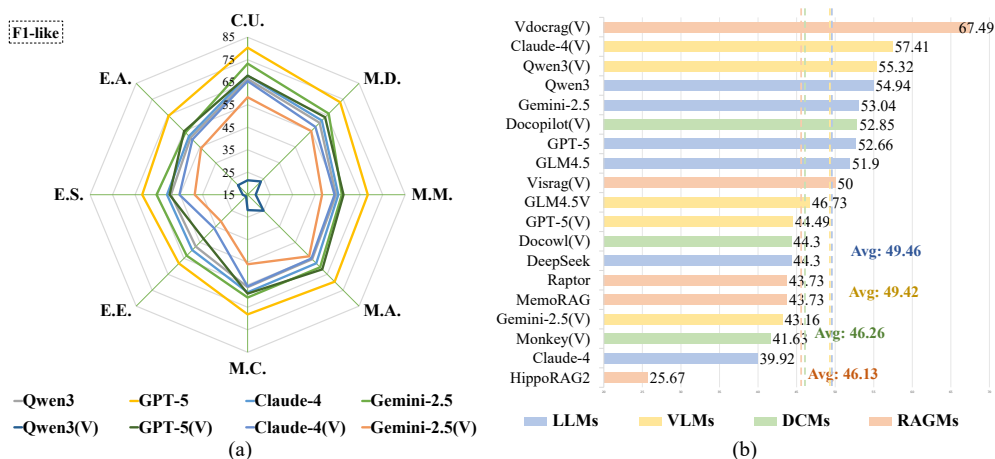


Figure 3: Comparison of LLMs and VLMs on open-ended question answering (F1-like score; left), and the performance of all models on claim verification tasks (ACC; right).

4.4 LLM JUDGMENTS VS. HUMAN ASSESSMENTS

To align LLM judgments with human assessments, we sampled 300 open-ended QA instances from the test set and generated predictions from all models in Section 4.1. Although prompts limited answers to 3,000 characters (Appendix B.1), actual output lengths varied widely. To control for length effects, for each QA instance, we selected the three model responses of most similar length and created pairwise comparisons for annotators to judge which answer is better in terms of correctness-completeness. During annotation, both presentation order and left-right placement were randomized, and model identities were masked to reduce bias.

Table 4: Agreement between evaluation configurations and human judgments (AUG: enhancement with title and abstract; SEP/JOI: separate/joint evaluation)

Setting	P-BT	S-BT	PW-AUC	Avg.
AUG+SEP	0.8955	0.9137	0.7125	0.8406
AUG+JOI	0.9003	0.9091	0.6966	<u>0.8353</u>
RAW+SEP	0.8773	0.9137	0.7054	0.8321
RAW+JOI	0.8759	0.9137	0.7100	0.8332

Table 5: Analysis of agreement between different LLM judges and human judgments.

Model	P-BT	S-BT	PW-AUC	Avg.
GPT-5	0.9213	0.9137	0.7255	<u>0.8535</u>
Gemini-2.5	0.9164	0.9137	0.7280	0.8527
GLM-4.5	0.8971	0.9091	0.7054	0.8372

Consistency Evaluation Metrics. We measure consistency between model judgments and human assessments using two metrics: **BT-based correlation (P-BT, S-BT)**, which applies the Bradley-Terry model (Turner & Firth, 2012) to convert pairwise outcomes into scores and correlates them with human preferences (Pearson/Spearman); and **pairwise AUC (PW-AUC)**, which directly compares model-predicted pairwise outcomes with human labels.

Prompt Configuration for LLM Judgments. We examine prompt design through an ablation of two factors: (1) whether to provide the title and abstract, and (2) whether to present evaluation metrics separately or jointly. Using GLM-4-Plus as the judge, we evaluated four configurations from their cross-combination. As shown in Table 4, including the title and abstract helps the judge understand the question’s context, while assessing each dimension independently mitigates error propagation that can arise when an anomalous score affects multiple dimensions in joint evaluation.

Which LLMs to Judge? We first used GLM-4-Plus to score 300 sampled QA instances under the configurations above. From these results, we identified the top three models (GPT, Gemini, GLM) and measured their alignment with human assessments (Table 5). The two models with the highest alignment were then jointly chosen as evaluation judges to reduce single-judge bias. This procedure is interpretable, extensible, and adaptable to other tasks as resources allow.

4.5 CASE STUDY

We conducted case studies across four key dimensions: (1) common failure modes of current models (Section 4.5.1), (2) detailed analysis for each category (Section 4.5.2), (3) textual versus visual input (Appendix B.4.3), and (4) bottlenecks of RAG methods (Appendix B.4.4)

4.5.1 COMMON FAILURE MODES

This section illustrates common failure modes of current models in research paper comprehension. Additional examples are provided in Appendix B.4.1.

Example 1 (Degenerative Output Patterns): The model’s decoding collapses into uninformative content, highlighting the importance of long-form generation tasks to stress-test stability.

Example 2 (Necessity of Multimodal Grounding): A text-only model extracts conclusions from text, while a multimodal model grounds claims in visual evidence (e.g., watermarks in Figure 10), revealing cross-modal reasoning capabilities beyond text-only evaluation.

Example 1	Example 2
<p>Paper: NaturalSpeech 2: Latent Diffusion Models are Natural and Zero-Shot Speech and Singing Synthesizers</p> <p>Question: What are the roles of SoundStream and WaveNet ... and how do they interact with other components such as the prior model, diffusion model, and audio codec?</p> <p>Answer: SoundStream is used as the neural audio codec ... WaveNet is chosen as the architecture for the diffusion model ... The system consists of three components: a prior model, a diffusion model, and an audio codec...</p>	<p>Paper: AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning</p> <p>Question: What are the visualization results when the domain adapter training (stage 1) is entirely omitted from the process...?</p> <p>Answer: The visualizations show that when the domain adapter is entirely removed from the training pipeline, the watermarks from the training dataset appear on the synthetic animations... These results are presented in Appendix B.2 and Fig. 10.</p>
<p>VdocRAG(V): ...The audio codec and the diffusion model interact with the audio decoder to obtain the audio waveform. The audio codec ... audio waveform. [The previous sentence is repeated over 100 times, resulting in a 19,867-character response].</p>	<p>GPT-5: Omitting domain adapter training ... This occurs because visual appearance and motion are entangled... (Appendix B.2, Fig. 10, 1st row).</p> <p>GPT-5(V): When the domain adapter training (stage 1) is omitted entirely, ... resulting in artifacts such as visible watermarks transferring into the generated visuals (Figure 10).</p>

Figure 4: Representative case study from the RPC-Bench test set

4.5.2 DETAILED ANALYSIS OF EACH CATEGORY

Our evaluation across eight open-ended QA categories reveals clear differences. Detailed examples are provided in Appendix B.4.2.

Concept Understanding & Method Disambiguation: Require accurate in-depth understanding. High-capacity LLMs integrate contextual details effectively. VLMs often omit fine-grained elements, DCMs’ answers tend to be too general. RAG excels at information extraction but lacks summarization and reasoning capabilities.

Method Mechanics & Comparison: LLMs provide clear explanatory contrasts. RAG provides basic comprehension but offers less interpretive depth. VLMs may miss details, DCMs risk repetitive or partial answers.

Motivation Analysis & Experimental Setup: Focus on understanding the rationale behind methodological/experimental choices. LLMs and strong VLMs link design choices to their underlying motivation, DCMs and RAG often miss underlying rationale.

Experiment Exposition & Analysis: Demand retrieval from dispersed sources, sometimes with quantitative evidence. LLMs can perform qualitative or quantitative analysis of experimental results based on a research paper, VLMs further perform analyses via figure interpretation, RAG ensures detail fidelity. DCMs’ answers tend to be too general.

5 CONCLUSION

To comprehensively evaluate models’ ability to understand research papers, we introduce RPC-Bench, a large-scale benchmark comprising 4,050 papers and 46.3K QA pairs across nine categories, supporting both text and rendered page inputs. We develop an LLM–human collaborative annotation framework to ensure scalability and quality. The evaluation design addresses three key questions—whether to provide titles/abstracts as context, whether to assess metrics independently, and how to select the judging LLM—aiming to better align LLM judgments with human assessments. Our scoring protocol evaluates correctness, completeness, and conciseness. Experiments on 19 state-of-the-art models highlight persistent challenges, including limited use of multimodal information, insufficient conciseness, and weak reasoning over visual content. We envision RPC-Bench as a foundation for evaluating deep understanding and reasoning in large foundation models.

ETHICS STATEMENT

We confirm that this work adheres to the ICLR Code of Ethics and applicable standards of research integrity. The benchmark introduced in this study (RPC-Bench) was constructed exclusively from publicly available papers and their review–rebuttal pairs hosted on OpenReview. All data were

540 originally authored for public dissemination, and no private, confidential, or proprietary information
541 was included. No experiments in this work involve human subjects, personal health data, or sensitive
542 demographic attributes.

543 To ensure fairness, we performed quality control on all collected data, including removal of
544 low-quality or irrelevant content, and balanced sampling to avoid systematic bias in sources. The
545 proposed benchmark does not include any potentially harmful insights, discriminatory content, or
546 security-relevant information. All licenses and usage terms of the source data have been respected,
547 and dataset release will comply with the original terms of access.

548 Potential conflicts of interest, such as affiliations or sponsorships, have been disclosed in accordance
549 with conference policies. This study aims to advance model evaluation methodology for academic
550 paper comprehension without promoting or enabling malicious applications.
551

552 REPRODUCIBILITY STATEMENT

553 We have made every effort to ensure that our work is fully reproducible. Full details of the dataset
554 construction process, taxonomy design, annotation procedure, quality control steps, and evaluation
555 framework are described in Section 3 of the main text and further expanded in the Appendix. An
556 anonymous repository containing the PRC dataset (with train/validation/test splits), complete an-
557 notation and filtering scripts, and evaluation code for both binary verification and open-ended QA
558 tasks is provided in the main text. Hyperparameter settings and experimental configurations for all
559 evaluated models are also included.

560 All datasets and source code will be made publicly available upon publication under an open-access
561 license. These resources allow researchers to reproduce the dataset construction, annotation
562 pipeline, and evaluation results reported in this work.

563 REFERENCES

- 564 Anthropic. Introducing claude 4. <https://www.anthropic.com/news/claude-4>, 2025.
565 Accessed: 2025-05-23.
- 566 Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge,
567 Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu,
568 Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi
569 Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng
570 Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi
571 Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang
572 Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023. URL
573 <https://arxiv.org/abs/2309.16609>.
- 574 Tim Baumgärtner, Ted Briscoe, and Iryna Gurevych. Peerqa: A scientific question answering dataset
575 from peer reviews. *arXiv preprint arXiv:2502.13668*, 2025.
- 576 Franck Cappello, Sandeep Madireddy, Robert Underwood, Neil Getty, Nicholas Lee-Ping Chia,
577 Nesar Ramachandra, Josh Nguyen, Murat Keceli, Tanwi Mallick, Zilinghan Li, et al. Eaira: Es-
578 tablishing a methodology for evaluating ai models as scientific research assistants. *arXiv preprint
579 arXiv:2502.20309*, 2025.
- 580 Yuan Chang, Ziyue Li, Hengyuan Zhang, Yuanbo Kong, Yanru Wu, Zhijiang Guo, and Ngai Wong.
581 Treereview: A dynamic tree of questions framework for deep and efficient llm-based scientific
582 peer review. *CoRR*, abs/2506.07642, 2025. doi: 10.48550/ARXIV.2506.07642. URL <https://doi.org/10.48550/arXiv.2506.07642>.
- 583 Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit
584 Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the
585 frontier with advanced reasoning, multimodality, long context, and next generation agentic capa-
586 bilities. *arXiv preprint arXiv:2507.06261*, 2025.

- 594 Chao Deng, Jiale Yuan, Pi Bu, Peijie Wang, Zhong-Zhi Li, Jian Xu, Xiao-Hui Li, Yuan Gao, Jun
595 Song, Bo Zheng, et al. Longdocurl: a comprehensive multimodal long document benchmark
596 integrating understanding, reasoning, and locating. *arXiv preprint arXiv:2412.18424*, 2024.
597
- 598 Zehang Deng, Wanlun Ma, Qing-Long Han, Wei Zhou, Xiaogang Zhu, Sheng Wen, and Yang Xi-
599 ang. Exploring deepseek: A survey on advances, applications, challenges and future directions.
600 *IEEE/CAA Journal of Automatica Sinica*, 12(5):872–893, 2025.
- 601 Michael Desmond, Zahra Ashktorab, Werner Geyer, Elizabeth M Daly, Martin Santillan Cooper,
602 Qian Pan, Rahul Nair, Nico Wagner, and Tejaswini Pedapati. Evalassist: Llm-as-a-judge simpli-
603 fied. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 29637–
604 29639, 2025.
- 605 Jennifer D’Souza, Hamed Babaei Giglou, and Quentin Münch. Yescieval: Robust llm-as-a-judge
606 for scientific question answering. *arXiv preprint arXiv:2505.14279*, 2025.
607
- 608 Yuchen Duan, Zhe Chen, Yusong Hu, Weiyun Wang, Shenglong Ye, Botian Shi, Lewei Lu, Qibin
609 Hou, Tong Lu, Hongsheng Li, et al. Docopilot: Improving multimodal models for document-level
610 understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp.
611 4026–4037, 2025.
- 612 Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Tru-
613 itt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global:
614 A graph rag approach to query-focused summarization, 2025. URL [https://arxiv.org/
615 abs/2404.16130](https://arxiv.org/abs/2404.16130).
- 616 Team GLM, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego
617 Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang,
618 Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Jingyu Sun, Juanzi Li, Lei Zhao, Lindong Wu, Lucen
619 Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan
620 Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan
621 Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang,
622 Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi,
623 Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of
624 large language models from glm-130b to glm-4 all tools, 2024. URL [https://arxiv.org/
625 abs/2406.12793](https://arxiv.org/abs/2406.12793).
- 626 Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory:
627 Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*,
628 2025.
- 629 Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and
630 Jingren Zhou. mplug-docowl2: High-resolution compressing for ocr-free multi-page document
631 understanding, 2024. URL <https://arxiv.org/abs/2409.03420>.
- 632 Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao,
633 Zhao Yang, and Ji-Rong Wen. Flashrag: A modular toolkit for efficient retrieval-augmented
634 generation research. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 737–
635 740, 2025.
- 636
- 637 Maikel Leon. Gpt-5 and open-weight large language models: Advances in reasoning, transparency,
638 and control. *Information Systems*, pp. 102620, 2025.
- 639 Haitao Li, Junjie Chen, Qingyao Ai, Zhumin Chu, Yujia Zhou, Qian Dong, and Yiqun Liu. Cal-
640 ibraeval: Calibrating prediction distribution to mitigate selection bias in llms-as-judges. *arXiv
641 preprint arXiv:2410.15393*, 2024a.
- 642
- 643 Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multi-
644 modal arxiv: A dataset for improving scientific comprehension of large vision-language models.
645 *arXiv preprint arXiv:2403.00231*, 2024b.
- 646
- 647 Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and
Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal
models, 2024c. URL <https://arxiv.org/abs/2311.06607>.

- 648 Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. Spiq: A dataset for mul-
649 timodal question answering on scientific papers. *Advances in Neural Information Processing*
650 *Systems*, 37:118807–118833, 2024.
- 651
- 652 Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun
653 Huang. Memorag: Boosting long context processing with global memory-enhanced retrieval
654 augmentation. In *Proceedings of the ACM on Web Conference 2025*, pp. 2366–2377, 2025.
- 655
- 656 Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Man-
657 ning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth Inter-
658 national Conference on Learning Representations*, 2024.
- 659
- 660 Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu,
661 Michael Moor, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using llm agents as research
662 assistants, 2025. URL <https://arxiv.org/abs/2501.04227>.
- 663
- 664 Ryota Tanaka, Taichi Iki, Taku Hasegawa, Kyosuke Nishida, Kuniko Saito, and Jun Suzuki.
665 Vdocrag: Retrieval-augmented generation over visually-rich documents. In *Proceedings of the
666 Computer Vision and Pattern Recognition Conference*, pp. 24827–24837, 2025.
- 667
- 668 Heather Turner and David Firth. Bradley-terry models in r: the bradleyterry2 package. *Journal of
669 statistical software*, 48:1–21, 2012.
- 670
- 671 Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong
672 Pei, Armineh Nourbakhsh, and Xiaomo Liu. Docllm: A layout-aware generative language model
673 for multimodal document understanding. *arXiv preprint arXiv:2401.00908*, 2023.
- 674
- 675 Renqiu Xia, Song Mao, Xiangchao Yan, Hongbin Zhou, Bo Zhang, Haoyang Peng, Jiahao Pi,
676 Daocheng Fu, Wenjie Wu, Hancheng Ye, et al. Docgenome: An open large-scale scientific doc-
677 ument benchmark for training and testing multi-modal large language models. *arXiv preprint
678 arXiv:2406.11633*, 2024.
- 679
- 680 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang
681 Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu,
682 Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin
683 Yang, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang,
684 Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui
685 Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang
686 Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger
687 Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan
688 Qiu. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.
- 689
- 690 Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang,
691 Xu Han, Zhiyuan Liu, et al. Visrag: Vision-based retrieval-augmented generation on multi-
692 modality documents. *arXiv preprint arXiv:2410.10594*, 2024.
- 693
- 694 Yaohui Zhang, Haijing Zhang, Wenlong Ji, Tianyu Hua, Nick Haber, Hancheng Cao, and Weixin
695 Liang. From replication to redesign: Exploring pairwise comparisons for llm-based peer review.
696 *CoRR*, abs/2506.11343, 2025. doi: 10.48550/ARXIV.2506.11343. URL [https://doi.org/
697 10.48550/arXiv.2506.11343](https://doi.org/10.48550/arXiv.2506.11343).

698 A USE OF LARGE LANGUAGE MODELS (LLMs)

699 In this work, large language models (LLMs) were used solely as general-purpose assistive tools for
700 language refinement. Specifically, we employed LLMs to polish the phrasing and improve gram-
701 matical correctness in the manuscript. No part of the research design, idea generation, data analysis,
experimental execution, or substantive technical writing was performed by LLMs. All conceptual
contributions, scientific content, and experiments were conceived, implemented, and verified en-
tirely by the authors.

B SUPPLEMENTARY EXPERIMENTS

B.1 RESPONSE LENGTH ANALYSIS

To better understand output length characteristics, we analyzed the distribution of response lengths for all models, as summarized in Table 6. However, most models tended to produce responses approaching the upper bound, which we attribute to their limited ability to comprehend and reason over the research paper content, leading to verbose rather than concise answers. Notably, some models (such as DocOwl2, VdocRAG, and VisRAG) exhibited abnormally high maximum output lengths. Manual inspection revealed that these overly long outputs often consisted of repetitive, non-informative text generated when the model failed to answer the question effectively. Conversely, certain models registered a minimum response length of zero, indicating empty answers either due to refusal triggered by safety policies (e.g., GLM-4.5V) or an inability to provide a response. Overall, the most models struggle to effectively achieve content comprehension, information compression, and logical reasoning within the task constraints, revealing a fundamental gap between the demands of accurate, concise, and contextually grounded scholarly reasoning and the current capabilities of state-of-the-art systems.

Table 6: Response Length Analysis.

Model Type	Model	Length (char)		
		Avg	Max.	Min
LLM	DeepSeek-V3.1	1494.19	4350	51
	GLM-4.5	2127.67	10285	478
	Qwen3	1971.16	5527	289
	GPT-5	1717.14	4065	37
	Claude-4	2109.62	5198	85
	Gemini-2.5	1741.91	3985	330
DCM	DocOwl2	525.52	23251	0
	Docopilot	1108.01	4031	59
	Monkey	425.10	7351	10
VLM	GLM-4.5V	1090.89	4079	0
	Qwen3(V)	2019.61	7627	249
	GPT-5(V)	1332.63	4698	24
	Claude-4(V)	3530.55	9853	821
	Gemini-2.5(V)	1667.97	3256	274
RAG	HippoRAG2	1109.48	3011	39
	MemoRAG	594.26	1957	1
	Raptor	838.01	1956	0
	VdocRAG	2391.54	30179	1
	VisRAG	1232.17	13046	2
-	Ground True	829.63	5194	5

B.2 FINETUNE LLM ANALYSIS

We fine-tuned Qwen and LLaMA on the PRC training set, with results summarized in Table 7. Both models achieved consistent improvements in the overall Info. metric, increasing by 11.38% and 10.64%, respectively. Notably, conciseness improved significantly, whereas the F1-like score remained relatively stable. This suggests that, compared to correctness and completeness, models more readily learn to produce concise responses. In contrast, achieving high correctness and completeness imposes greater demands on the models' fundamental comprehension and reasoning capabilities.

Table 7: Performance Comparison in Fine-Tuning Experiments.

Model Type	Conc.	Corr.	Compl.	F1-like	Info.
Llama-3.1-8B-Instruct	41.56	34.75	30.92	32.72	13.60
Llama-3.1-8B-Instruct-FT	77.07	36.20	29.34	32.41	24.98
Qwen3-8B	48.20	38.53	32.52	35.27	17.00
Qwen3-8B-FT	78.58	39.30	31.82	35.17	27.64

B.3 EVALUATION OF MODEL CONCISENESS ACROSS TAXONOMY-DEFINED QUESTION TYPES

Figure 5 illustrates the performance of different models across various question types. Most models show minimal variation in conciseness scores across question categories, forming an almost concentric pattern in the radar chart, with no score exceeding 65%. This underscores the difficulty models face in generating responses that are both relevant and precise. Across all question categories, text-based inputs generally yield more concise outputs than image-based inputs. We attribute this to the models’ weaker capability in interpreting visual inputs, where the relative loss of explicit textual detail may result in responses that convey less relevant information while being unnecessarily verbose. This pattern is evident in the performance gap between text- and image-based inputs for Qwen3, Gemini-2.5, and Claude-4. An exception is GPT-5(V), which achieves the highest answer conciseness, even surpassing its text-based variant. This result suggests that GPT-5(V) can more effectively leverage visual information, consistent with its strong F1-Like score (see Figure 3 (left)).

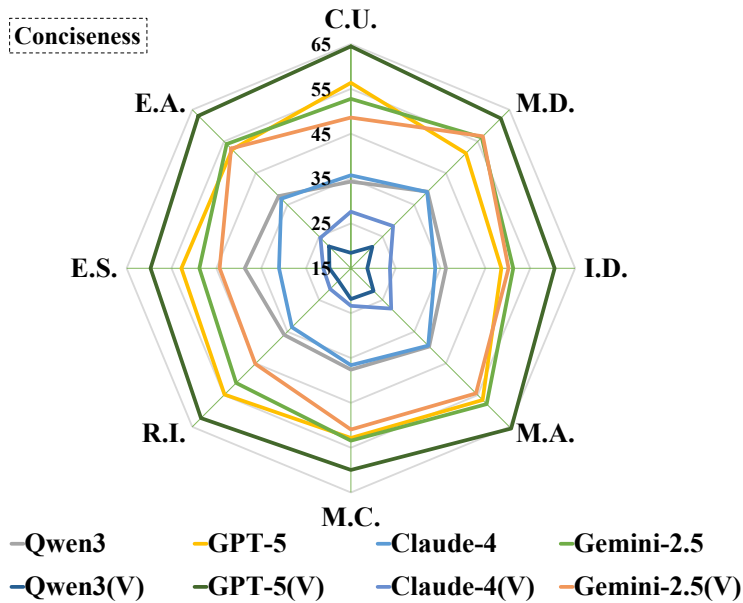


Figure 5: Model Conciseness across Taxonomy-Defined Question Types

B.4 MORE CASE STUDY

B.4.1 COMMON FAILURE MODES

Below are two additional case studies:

Example 3 (Hallucination): The model sometimes wrongly denies information that is actually present in the source document, highlighting the need for tasks that test precise data extraction and catch such factual-verification errors.

Example 4 (Precise Output Failures): Despite the prompt explicitly constraining the output format to strict booleans (True/False), both models violate this requirement: one returns a self-contradictory invalid answer, while the other appends extraneous characters.

Example 3	Example 4
<p>Paper: Safe RLHF: Safe Reinforcement Learning from Human Feedback</p> <p>Question: What were the experimental results comparing the paper’s method to additional safety RLHF baselines such as Constitutional AI and Safety SFT?</p> <p>Answer: Win rates for three different methods rated by GPT-4: ... — Constitutional AI — Safety SFT — Helpfulness Win Rate — 71.8% — 40.2% — 53.6% — ... the experiments reveal interesting insights...</p>	<p>Paper: Eureka: Human-Level Reward Design via Coding Large Language Models</p> <p>Question: Does the EUREKA generate reward functions statistically significantly better than human-designed rewards, based on a Mann-Whitney U-Test with a p-value threshold of 0.05?</p> <p>Answer: True</p>
<p>Claude-4: ...the experimental comparison of Safe RLHF to Constitutional AI and Safety SFT baselines was very limited. The paper did not conduct comprehensive experiments comparing their method to Constitutional AI and Safety SFT.</p>	<p>Claude-4: False True</p> <p>Claude-4(V): True 1</p>

Figure 6: Additional Case study

B.4.2 DETAILED ANALYSIS OF EACH CATEGORY

Representative examples for each category are presented below.

Example 1: Concept Understanding
<p>Question: What is "cross-LoRA attention" as defined in the paper?</p> <p>Answer: Cross-LoRA attention is an attention mechanism that allows for combining concept and style LoRAs. It splits queries, keys, and values, such that queries from the concept branch attend to keys and values from both the concept and style branches, effectively merging their representations.</p> <p>GPT: Cross-LoRA attention is an attention mechanism designed to combine the concept and style LoRAs... It involves splitting the queries, keys, and values (Q, K, V) into concept and style components. The concept queries attend to the concatenated keys and values from both the concept and style branches...</p> <p>GPT(V): Cross-LoRA attention is the mechanism the authors propose to merge the learned representations of a concept LoRA and a style LoRA during inference... The key idea is to take the query from the concept branch and let it attend to the keys and values from both the concept and style branches.</p> <p>DocOwl2: Cross-LoRA attention is a novel method that enables the model to focus on different aspects of the input image, allowing for more detailed and accurate image generation.</p> <p>VisRAG: Cross-LoRA attention is defined as a mechanism for merging LoRA weights by splitting the query (q) into a concept part (qc) and a style part (qs), and then having the concept query attend to keys and values from both branches.</p>
Example 2: Method Disambiguation
<p>Question: How does the positional embedding used in Rectified Flow Transformers (RFTs) differ from the one used in Diffusion Transformers (DiTs)?</p> <p>Answer: RFTs condition on a continuous time variable 't' from '[0, 1]', while DiTs condition on a discrete noise level 'Σ'. This allows RFTs to leverage the properties of rectified flow for training.</p> <p>Claude-4: The key difference is that RFTs condition on the continuous time variable 't' which ranges from 0 to 1... In contrast, DiTs condition on discrete noise levels 'Σ'.</p> <p>Qwen3(V): Both RFTs and DiTs utilize positional embeddings... However, the paper does not explicitly detail the structural differences... it focuses on the Rectified Flow method itself rather than a component-wise comparison of embeddings.</p> <p>DocOwl2: The positional embedding in RFTs is based on the transformer architecture, while the positional embedding in DiTs is based on the diffusion model architecture.</p>

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

HippoRAG2: Rectified Flow Transformers (RFT) condition on time ‘ t ’, whereas DiTs condition on the noise level ‘ Σ ’.

Example 3: Method Mechanics

Question: How is normalization applied in the description of Classification Accuracy?

Answer: Normalization is performed over the multi-choice options given, rather than over all sequences in the vocabulary.

Claude-4: The key aspect of normalization is that it is ”performed over the options given, rather than over all sequences.” This means that instead of normalizing probabilities across the entire vocabulary... the normalization is restricted to only the specific answer choices provided...

Claude-4(V): ...the model uses the probability of the full token sequences corresponding to each class. These probabilities are renormalized to sum to 1 over the provided options for the task... It is done **over the given options only**, rather than over all possible sequences in general...

Monkey: Normalization is applied... by dividing the number of correctly classified instances by the total number of instances in the test set.

MemoRAG: Normalization is done over the options given, rather than over all sequences.

Example 4: Motivation Analysis

Question: Why was RAFT not tested on tasks such as text-to-SQL or reasoning-based QA, given its potential effectiveness in those domains?

Answer: RAFT is specifically designed to enhance the base LLM’s performance in domain-specific RAG settings. While it might improve performance in coding or reasoning domains, studying RAFT’s performance on those tasks falls outside the scope of this paper, as it focuses on creating domain expert RAG models rather than general post-training strategies.

Claude-4: ...RAFT was not tested on text-to-SQL or reasoning-based QA tasks because the authors made specific dataset choices aligned with their research objectives and scope limitations... The paper explicitly states their dataset selection criteria...

Claude-4(V): ...due to the focus and scope of this paper, which targets domain-specific open-book question answering (QA) settings... RAFT’s method depends on training with **golden and distractor documents**... [This] is fundamentally different from generating executable code (SQL queries)...

Monkey: The paper does not provide a direct reason...

MemoRAG: RAFT was not tested on... text-to-SQL or reasoning-based QA because the paper’s focus is specifically on adapting pre-trained LLMs to domain-specific retrieval-Augmented Generation (RAG) in specialized domains.

Example 5: Method Comparison

Question: How does Self-RAG differ from other RAG methods, such as those that use iterative retrieval or reranking?

Answer: Self-RAG differs by adaptively retrieving passages on-the-fly and using self-reflection. Unlike iterative retrieval, it retrieves when needed during generation. Unlike reranking, which happens before generation, Self-RAG’s critiques happen post-generation to select the best output. It generates ”reflection tokens” to make these decisions.

Gemini-2.5: Self-RAG differs by training a single language model to adaptively retrieve passages on an on-demand basis and to **self-reflect on the retrieved passages**... Unlike iterative retrieval which retrieves a fixed number of documents... Self-RAG retrieves passages whenever it deems necessary... Unlike reranking... Self-RAG uses its own critique mechanism after generation...

918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

Gemini-2.5(V): Self-RAG differs in that it is an on-demand retrieval system that works during the decoding process... It also generates special tokens ('[Retrieve]', '[Critique]') that allow it to assess its own generation, unlike other methods that re-rank at the start.

Monkey: Traditional RAG methods retrieve information, then generate a response. Self-RAG adds a "critique" step where the model evaluates its own generation. However, the paper does not compare it to iterative retrieval.

HippoRAG2: Self-RAG is different because it trains one arbitrary LM to generate text and reflect on its own output... by generating special tokens... It retrieves passages one at a time and in an adaptive manner, unlike conventional RAG that retrieves a fixed number of passages.

Example 6: Experimental Exposition

Question: What are the effects of not including a weight threshold, and are there specific tasks where this negatively impacts the results?

Answer: An ablation study showed minimal impact on most tasks. However, three specific tasks—'Date Understanding,' 'Disambiguation,' and 'Hyperbaton'—showed notable performance declines, with an average decrease of 1.2%.

GPT-5: Effect on overall performance: Removing the weight threshold... has minimal impact on most tasks... reduces the average performance across BBH from 34.7 to 33.5... Tasks negatively impacted: Three tasks show notable drops... Date Understanding: 32.9 → 17.7... Disambiguation: 45.2 → 40.6... Hyperbaton: 62.8 → 55.5...

GPT-5(V): The paper does not study or report a "weight threshold" mechanism.**... Consequently: - The effects of not using a weight threshold are not evaluated or quantified in the paper. - No tasks are identified where omitting a weight threshold harms performance.

DocOwl2: ...including a weight threshold does not significantly affect the method's performance, and there are no specific tasks where this adjustment negatively impacts the results.

VisRAG: The removal of the threshold had minimal impact on the majority of tasks... However, three tasks namely Date Understanding, Disambiguation and Hyperbaton exhibited notable effect... average decrease of 1.2%...

Example 7: Experimental Setup

Question: Can the success/fitness function be used to initialize the Eureka reward search process?

Answer: Yes, the success/fitness function can be used. Section 4.4 describes an experiment where a human - supplied reward function was used to initialize the process.

Gemini-2.5: Yes, the success/fitness function can be used to initialize the Eureka reward search process, although it is not the default procedure... Section 4.4 explains that initializing the search with an existing reward function "requires no modification to EUREKA."

Gemini-2.5(V): Based on the provided research paper, the success/fitness function is not used to initialize the Eureka reward search process... The initialization of the reward search process involves providing the Large Language Model (LLM) with two key pieces of information as context: 1. The environment source code. 2. A natural language task description.

Monkey: Yes, the success/fitness function can be used to initialize the Eureka reward search process.

MemoRAG: Yes, the success/fitness function can be used to initialize the Eureka reward search process.

Example 8: Experimental Analysis

Question: What are the major errors identified in the error analysis conducted in Section 5.1, and how do these errors manifest in the model’s predictions?

Answer: The major errors identified in the error analysis lie within the click region and scroll direction predictions. The model tends to click the wrong place or scroll in the wrong direction, despite predicting the right action most of the time. This suggests a need for improving the model’s ability to understand screen layouts.

GPT-5: Section 5.1 identifies click region localization and scroll direction prediction as the major error sources... These errors manifest as the model selecting the correct type of action but failing at the low - level execution...

GPT-5(V): Section 5.1 shows that Auto - UI’s main errors occur at the execution level, not in deciding what to do. Error types: inaccurate click localization and incorrect scroll direction... Evidence: averaged category accuracies are 67.4% for click region and 82.0% for scroll direction...

DocOwl2: The model’s predictions are not accurate because of overfitting and underfitting... identified as model drift and model error...

MemoRAG: Major errors include incorrect location prediction during clicks (8%) and misdirection during scroll movements (4%).

B.4.3 TEXTUAL VS. VISUAL INPUT

This section provides a qualitative comparison of model behavior under purely textual and visual inputs.

Example 1, 2, and 4 shows that LLMs consistently deliver higher factual accuracy, greater answer completeness, and closer alignment with reference answers than VLMs. In Example 3, VLMs accurately identified methodological boundaries (e.g., excluding tasks outside the intended domain-specific RAG scope) and maintained reasoning consistency with the reference. This indicates that VLMs can effectively capture high-level conceptual scope, especially when the task focuses on boundary recognition rather than detailed fact retrieval.

Overall, LLMs excel in tasks requiring precise, evidence-linked, detail-rich answers from textual content. VLMs occasionally match or surpass LLMs in identifying conceptual scope from visual inputs. The observed gap largely reflects VLMs’ difficulty in extracting high-precision information from PDF-rendered images of dense academic content (e.g., complex tables, multi-column layouts, long texts), which often results in incomplete or noisy answers compared with clean text inputs.

Example 1

Question: Can the success/fitness function be used to initialize the Eureka reward search process?

Answer: Yes, the success/fitness function can be used. Section 4.4 describes an experiment where a human-supplied reward function was used to initialize the process.

GPT-5: Yes. EUREKA can be initialized from any provided reward function by substituting it for the first iteration’s reward, without modifying the algorithm. The paper demonstrates this “initialization” mechanism with human-written rewards: “we can simply substitute the raw human reward function as the output of the first EUREKA iteration” (Sec. 4.4; App. D.2). In the benchmarks, the success/fitness function F is provided as executable reward code and is used directly as the “Sparse” reward baseline (“These are identical to the fitness functions $F...$,” Sec. 4.1; App. B). Therefore, F can be used in exactly the same way as an initialization for EUREKA: set the first iteration’s reward to F , evaluate it, and then let EUREKA’s evolutionary search and reward reflection refine it in subsequent iterations (Alg. 1; Sec. 3.2–3.3).

GPT-5(V): Yes, the success/fitness function (F) can be used to initialize the Eureka reward search process, as outlined in the paper. Eureka explicitly uses (F) to evaluate the quality of candidate reward functions generated during its iterative optimization process (Section 3.2). (F) measures how effectively a policy derived from a reward function (R) solves a task (Definition 2.1). In the

initial phase, the fitness function is employed to assess the performance of reward functions sampled zero-shot from the coding LLM, enabling an informed starting point for the evolutionary search process (Section 3.1). While the initial rewards may exhibit suboptimality relative to the fitness metric, this evaluation ensures that the process begins with executable and task-relevant reward functions, which are progressively refined in subsequent iterations using evolutionary search and reward reflection. Thus, (F) plays a critical foundational role in initializing and guiding Eureka’s reward optimization.

Example 2

Question: What are the details of LoRA fine-tuning, including the rank, learning rate, and model architecture used?

Answer: The details of LoRA fine-tuning include incorporating trainable linear low-rank modules into the query and value projections of each self-attention layer, following Hu et al. (2021). The rank of LoRA is set to 32, the learning rate is $1e - 4$, and the model architecture is the Gecko PaLM model.

GPT-5: LoRA fine-tuning details (as used in SQL-PaLM): - Placement: Trainable low-rank adapters inserted into the query and value projections of each self-attention layer. - Rank: 32.- Learning rate: $1e - 4$. - Model architecture: PaLM-2 Gecko (Gecko PaLM model).

GPT-5(V): The LoRA (Low-Rank Adaptation) fine-tuning in the SQL-PaLM framework employs the following attributes: 1. Rank: LoRA fine-tuning utilizes a rank of 64, representing the dimensionality of the low-rank decomposition matrices used to adapt large model weights. 2. Learning Rate: A fixed learning rate of $5e-4$ is used for the training process with LoRA. 3. Model Architecture: LoRA is applied to the PaLM-2-Large model, a pretrained large-scale Transformer-based architecture. These details align with the authors’ exploration of parameter-efficient fine-tuning techniques tailored for scaling Text-to-SQL task performance.

Example 3

Question: Why was RAFT not tested on tasks such as text-to-SQL or reasoning-based QA, given its potential effectiveness in those domains?

Answer: RAFT is specifically designed to enhance the base LLM’s performance in domain-specific RAG settings. While it might improve performance in coding or reasoning domains, studying RAFT’s performance on those tasks falls outside the scope of this paper, as it focuses on creating domain expert RAG models rather than general post-training strategies.

GPT-5: Two points from the paper address this: - Reasoning-based QA: RAFT was evaluated on a reasoning-intensive QA benchmark. HotpotQA—an explainable, multi-hop QA dataset—is one of the core evaluations, and RAFT shows large gains there (e.g., 35.28 vs. much lower baselines in Table 1; further analysis in Sections 4.1–4.3). Thus, RAFT was tested on reasoning-based QA within the paper’s scope. - Text-to-SQL: The paper’s stated focus is domain-specific, document-centric RAG: “we only care about testing the LLM on the same set of documents,” and “the LLM will be used to respond to the questions, whose answers can be found within a collection of documents.” All methods, data construction, and evaluations are built around reading retrieved documents, citing verbatim spans, and being robust to distractor documents. The chosen datasets (Wikipedia-based QA, API documentation, PubMed) reflect this scope. Tasks like text-to-SQL are not part of this document-retrieval setting the paper studies, so they were outside the evaluation slate.

GPT-5(V): Because the paper’s goal and problem setting are narrowly defined as domain-specific open-book QA with retrieval. The method and experiments are built around: - Training and testing with a question plus a set of retrieved documents, where “answers can be found within a collection of documents” in the target domain (Domain-Specific Open-Book Exam section). - RAFT’s training format requires a golden document D^* that contains the answer and supports chain-of-thought with verbatim citations from the relevant document (Section 3). - The evaluation is designed to study reading from documents and robustness to distractors in RAG, using datasets such as PubMed, HotPotQA, and Gorilla APiBench that match this setting (Section 4). Tasks like text-to-SQL or pure reasoning-based QA do not align with this document-grounded RAG setup and do not provide the

golden supporting document and citations RAFT relies on. Hence, they were outside the scope of the study and were not evaluated.

Example 4

Question: How does WizardLM address instructions that are not included in the evolved set?

Answer: WizardLM addresses instructions not included in the evolved set through In-Breadth Evolving, which evolves entirely new instructions to enhance topic coverage, skill coverage, and overall diversity. This approach is demonstrated in Section 4.5 and Appendix J, showing improved diversity in the evolved instruction data compared to the original data.

GPT-5: WizardLM handles missing or failed evolutions in two ways: - During evolution: If an instruction fails to produce a valid evolved version (per the four failure checks), the evolved sample is discarded and the original instruction is put back into the pool “as is,” to be attempted again in subsequent evolution epochs. - For training data assembly: After all rounds, the final fine-tuning set is the merge of the initial instruction dataset and all successfully evolved instructions. Thus, instructions that never appear in the evolved set are still included via their original form. Additionally, WizardLM uses In-breadth Evolving to create brand-new, rarer instructions from existing ones, expanding topic and skill coverage to mitigate gaps where specific instructions are not represented in the evolved set.

GPT-5(V): WizardLM does not discard instructions that are not part of the evolved set. - During evolution, if an instruction fails to evolve (e.g., gives no information gain, is hard to answer, yields trivial output, or copies words from the prompt), it is “placed back as it is,” to be attempted again in later evolution rounds. - When building the final fine-tuning data, the authors merge the initial instruction dataset with all evolved instructions and shuffle them, ensuring unevolved original instructions are still included in training. Thus, instructions not included in the evolved set are retained and reused, both for further evolution attempts and by inclusion in the final fine-tuning corpus.

B.4.4 BOTTLENECKS OF RAG METHODS

We conducted a further analysis of the RAG-based approach, focusing on representative cases:

Example 1: The model successfully retrieved information related to the question but failed to use it effectively in answer generation. Key links between the paper’s context and the question were overlooked, and model-generated hallucinations were introduced.

Example 2: The model failed to retrieve the key information needed to answer the question, such as the main innovations of the method. Without this critical material, the generated answer was incomplete or incorrect.

Example 3: The model broke the question down into sub-questions, retrieved all necessary content, and produced a coherent, complete answer.

Overall, The main bottleneck of current RAG-based approach lies in their limited ability to accurately understand complex questions, retrieve targeted knowledge, and integrate information from multiple sources into a unified, meaningful answer.

Example 1

Question: Under what conditions does the statement $\mathcal{G}_g \geq 0$ hold, and what could lead to $\mathcal{G}_g < 0$?

Answer: The statement $\mathcal{G}_g \geq 0$ holds under the assumption that training and test data come from the same distribution. If this assumption is violated and the test data is simpler (i.e., from a distribution with lower entropy) than the training data, it can lead to $\mathcal{G}_g < 0$.

Retrieve: and the gap \mathcal{G}_r is a non-negative value

Note that $p_{\text{aug}}(\mathbf{x}') = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\text{train}}}[p_{\text{aug}}(\mathbf{x}' | \mathbf{x})]$ only extrapolates from individual data points $x \sim \mathcal{D}_{\text{train}}$ and has density outside the support of $p_{\text{data}}(\mathbf{x})$ (e

$\mathcal{G}_a = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\text{test}}}[\text{ELBO}_{\theta^*}(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\text{test}}}[\text{ELBO}_{\Theta}(\mathbf{x})]$
the ELBO on $\mathcal{D}_{\text{train}}$ is greater than or equal to the ELBO on $\mathcal{D}_{\text{test}}$

Therefore the ELBO on $\mathcal{D}_{\text{test}}$ can be higher than the ELBO on $\mathcal{D}_{\text{train}}$ if $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ are not drawn from the same distribution

and $\mathcal{D}_{\text{test}}$ has a lower entropy than $\mathcal{D}_{\text{train}}$
Hence, the continuity we gain by replacing $\mathcal{D}_{\text{train}}$ with $p_{\text{DM}}(\mathbf{x}')$ is not for free, we lose a small amount of information about $\mathcal{D}_{\text{train}}$

$$\mathcal{G}_r = \mathbb{E}_{\mathbf{x}^r \sim \mathcal{D}_{\text{test}}} \mathbb{E}_{\mathbf{x}^a \sim p(\mathbf{x}^a | \mathbf{x}^r)} [\text{MS-SSIM}(\mathbf{x}^r, \mathbf{x}^a)]$$

problem (ii) corresponds to density of $\bar{p}_{\text{aug}}(\mathbf{x}')$ outside of $p_{\text{data}}(\mathbf{x})$
For VAEs, we refer to the difference between training and test set ELBO as the generalization gap
Therefore, $\mathcal{G}_g \geq 0$ A smaller \mathcal{G}_g corresponds to a better generalization performance of a VAE Remark
(Test data entropy can also affect the ELBO value)

Output: ... ($\mathcal{G}_g \geq 0$) holds when player (i) maintains sufficient SCs and units, avoiding disbandment; ($\mathcal{G}_g < 0$) occurs if player (i) has few SCs, disbands units, or is effectively blocked by player (-i).

Example 2

Question: What are the novel aspects of the proposed method compared to prior work such as Lee et al. (2019), and how do they contribute to the improvement in performance?

Answer: ... The improvement in performance over ICT mainly comes from three factors: using MoCo to handle negatives, which allows scaling to a large number of negatives; the sampling procedure to generate pairs of (query, key); and using data from both CC-net and Wikipedia for training. The paper will be updated to highlight the origin of these gains. ...

Retrieve: A downside of this approach is that it requires extremely large batch sizes to work well Chen et al (2020), with Qu et al (2021) reporting improvement in the context of information retrieval up to 8192 negatives This method has been widely used to train information retrieval models with supervised data Chen et al (2017b); Karpukhin et al (2020) and was also considered when using ICT to pre-train retrievers by Lee et al

Hence, we believe that most of the improvement compared to the state-of-the-art retrievers can be attributed to our contrastive pre-training strategy Table 7: Impact of data augmentations We report nDCG@10 without fine-tuning on MS MARCO

Third, when used as a pre-training method before fine-tuning on MS MARCO, our technique leads to strong performance on the BEIR benchmark We perform ablations to motivate our design choices, and show that cropping works better than the inverse Cloze task Finally we train a multilingual dense retriever with contrastive learning and show that it achieves state-of-the-art performance Code and pre-trained models are available here: <https://github.com/facebookresearch/contriever> 2 Related work It should be noted that our fine-tuning procedure on MS MARCO is simpler than for other retrievers, as we use a simple strategy for negative mining and do not use distillation Our model would probably also benefit from improvements proposed by these retrievers, but this is beyond the scope of this paper Finally, we illustrate the benefit of our retriever compared to BM25 in a few-shot setting, where we have access to a small number of in-domain retrieval examples

Interestingly, as shown in Table 7, the random cropping strategy outperforms the inverse cloze task in our setting We believe that random cropping, leading to the identical distributions of keys and queries, leads to more stable training with MoCo compared to ICT This might explain part of the difference of performance between the two methods We also investigate whether additional data perturbations, such as random word deletion or replacement, are beneficial for retrieval Training data

In this section, we briefly review relevant work in information retrieval, and application of machine learning to this problem This is not an exhaustive review, and we refer the reader to Manning et al (2008), Mitra et al (2018) and Lin et al (2020) for a more complete introduction to the field Term-frequency based information retrieval

The methods described in the previous paragraph were applied to re-rank documents, which were retrieved with a traditional IR system such as BM25 Gillick et al (2018) first studied whether continuous retrievers, based on bi-encoder neural models, could be viable alternative to re-ranking In the context of question answering, Karpukhin et al (2020) introduced a dense passage retriever (DPR) based on the bi-encoder architecture

We report results, without fine-tuning on MS MARCO in Table 6 We observe that the difference of performance between the two methods is small, especially after fine-tuning on MS MARCO We thus propose to use MoCo as our contrastive learning framework, since it scales to a larger number of negative examples without the need to increase the batch size Number of negative examples

Finally, we also consider additional data augmentations such as random word deletion, replacement or masking We use these perturbations in addition to random cropping 3 1 3 Building large set of negative

pairs An important aspect of contrastive learning is to sample a large set of negatives Most standard frameworks differ from each other in terms of how the negatives are handled, and we briefly describe two of them, in-batch negative sampling and MoCo, that we use in this work

arXiv preprint arXiv:2002.08909, 2020 3, 5, 7 Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick Momentum contrast for unsupervised visual representation learning In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 9729–9738, 2020

Output: The proposed method introduces a DRL-based heuristic that generates local moves via a deep policy network, learns longer-sighted search policies, utilizes a GNN-based representation for JSSP, and employs a message-passing mechanism for efficient solution evaluation, leading to improved performance over prior methods.

Example 3

Question: What is the purpose and methodology of the In-breadth Evolving analysis, and how does it demonstrate the diversity of the dataset compared to ShareGPT and Alpaca (Self-Instruct)?

Answer: The Analysis of In-breadth Evolving section aims to enhance topic coverage, skill coverage, and overall dataset diversity. The methodology involves using BERT to encode instructions into 768-dimensional embeddings, applying t-SNE to reduce the embeddings to 2 dimensions, and using k-means clustering to partition instructions into 20 clusters for visualization. As shown in Figure 7, the data points of the dataset are more dispersed than those of ShareGPT and Alpaca (Self-Instruct), indicating better topic diversity.

Retrieve: In-breadth Evolving aims to enhance topic coverage, skill coverage, and overall dataset diversity

Here is instruction of Example N Rewritten Prompt: In-Breadth Evolving aims to enhance topic coverage, skill coverage, and overall dataset diversity Open-domain instruction finetune datasets (e.g., Alpaca, ShareGPT, etc.) are typically small in scale, lacking topic and skill diversity To solve this problem, we designed a prompt to generate a completely new instruction based on the given instruction, requiring the new instruction to be more long-tailed

As shown in the Figure 7, the data points of our dataset are more dispersed than ShareGPT and Alpaca (Self-Instruct), which indicates the better topic diversity in our instructions [(images/3c88da3581f2a7f8ccd6105127fb98eef3bcb4ba3c4205b76aaad6ceef6e492a.jpg)]

The results indicate that (i) the ShareGPT is a better seed for evol-instruct than Alpaca, (ii) larger evolved data size can improve model capacity, and (iii) our proposed Evol-Instruct method is not dependent on ChatGPT, other strong open source model such as Llama-2 is also a good substitute for

, Llama-1 65B, Llama-2, Mistral-7B) indicate that our Evol-Instruct can be widely applied to various pre-trained models Analysis of In-depth Evolving The Figure 5a and 5b presents an ablation study investigating the impact of the number of data evolution rounds To study the depth of the evolving process, we use ChatGPT to judge the difficulty level of instruction The used prompt please refer to Appendix E

, 2023a), and Tulu (Wang et al., 2023) 4.2 EXPERIMENT DETAIL To construct the dataset, we initialize it with the 52k instruction dataset of Alpaca and iteratively perform M evolutions, where $M = 4$ For each instruction in each round of evolution, we randomly select one evolving prompt from total six prompts (i.e.

Table 2: WizardLM with different data seed, data size, evol model, and base model size [(images/0e2a6d6f9e7ed060e716f7bc3d0b1fc4001af54061952a7dcfec6a8a4f09a6e1.jpg)] Figure 5: The difficulty level between ShareGPT, Alpaca, and our four epochs of evolved instruction

In order to study the impact of different data seeds Evol models scale of evolved dataset pre-trained models on our proposed method we conducted the following experiments a) Using 70k ShareGPT as the seed data to obtain WizardLM-13b (ShareGPT Seed) b) Using LLaMA-2-70B-Chat to replace ChatGPT as the evolutionary execution model to obtain WizardLM-13b (LLaMA-2-70B-Chat Evol)

To examine (qualitative analysis) the breadth (diversity) of different dataset, we firstly use BERT to encode each instruction and get its embedding with 768 dimensions, then use a dimension reduction algorithm named t-SNE to reduce embedding dimension to 2, finally we apply a clustering algorithm k-means to partition the instructions of each dataset into 20 clusters for an intuitive visualization

Alpaca used a dataset of 50k instructions generated from a limited (e.g., 175 samples) seed set of manually-written instructions Our work is different from InstructGPT and Vicuna in that we use AI-generated data for instruction fine-tuning Unlike Alpaca’s self-instruct Wang et al (2022a) generation method, Evol-Instruct can control the difficulty and complexity level of the generated instructions

Output: The purpose of the In-breadth Evolving analysis is to enhance topic coverage, skill coverage, and overall dataset diversity. The methodology involves encoding each instruction using BERT to obtain 768-dimensional embeddings, reducing the dimensions to 2 using the t-SNE algorithm, and then applying the k -means clustering algorithm to partition the instructions into 20 clusters for visualization. The analysis demonstrates the diversity of the dataset by showing that the data points in the cluster scatter plot are more dispersed compared to ShareGPT and Alpaca (Self-Instruct), indicating better topic diversity in the instructions.

C BENCHMARK DETAILS

C.1 QUALITY-BASED FILTERING CRITERIA

We perform quality-based filtering at two distinct stages (comment–response pairs and QA items) to remove low-quality items that cannot be reliably answered using only the paper’s content. The filtering criteria are:

- Temporary or editorial issue: corrections of grammar/spelling errors (e.g., “We corrected ‘benchamrks’ to ‘benchmarks’”), figure color/font adjustments, formatting changes or adding references, open-sourcing code/data (e.g., “Added reference to Smith et al.”), where the response merely acknowledges the fix without academic substance.
- External resource dependency: responses whose validity depends on external materials not contained in the paper (e.g., “More cases: <https://...>”), or indirect or evasive replies (e.g., “See Section X”).
- Non-substantive commitments: promises of future additions (e.g., “We will add a limitations section”, “Will address in future work”) without providing specific details or a concrete resolution in the current submission.

C.2 ANNOTATION PLATFORM

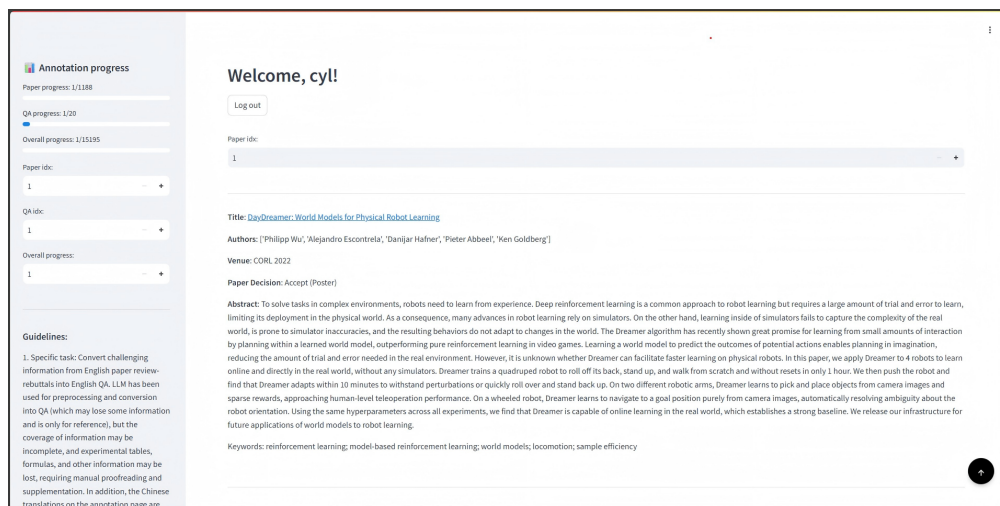


Figure 7: Screenshot of the Annotation Interface 1

C.3 REVIEW PLATFORM

C.4 DECOMPOSE PROMPT

1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349

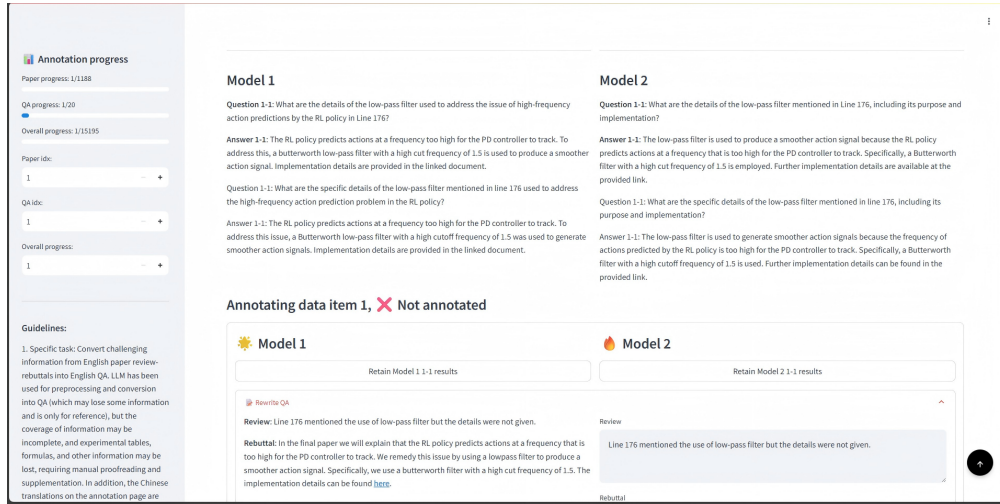


Figure 8: Screenshot of the Annotation Interface 2

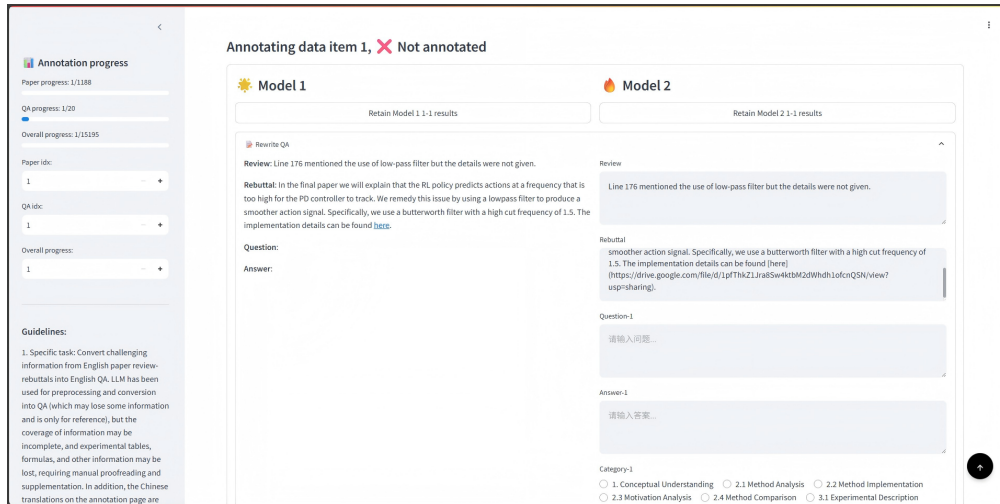


Figure 9: Screenshot of the Annotation Interface 3

You are an excellent reviewer of papers. You are tasked with extracting QA pairs from the "review", "rebuttal" and "extra_rebuttal" sections of a conference paper submission. This process includes identifying "review" provided by reviewers and pairing them with the corresponding answers authored by the paper's authors, utilizing content from both the "rebuttal" and any relevant "extra_rebuttal" sections. Your goals are: Extract and classify the QA pairs. Ensure that references and citations in the rebuttal are preserved in their original format within the answers, maintaining the academic rigor and clarity. Determine whether each question-answer pair is 'multimodal-related,' a broad concept that includes questions explicitly about the figures and tables in the paper or questions that can only be answered by referring to the contents of these figures and tables.

Input Structure:

review: Concatenation of all reviews, including multifaceted evaluations of the paper and any responses or questions directed at the authors' rebuttal.

rebuttal: The content in the rebuttal is a concatenation of the answers to all the review questions.

1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403

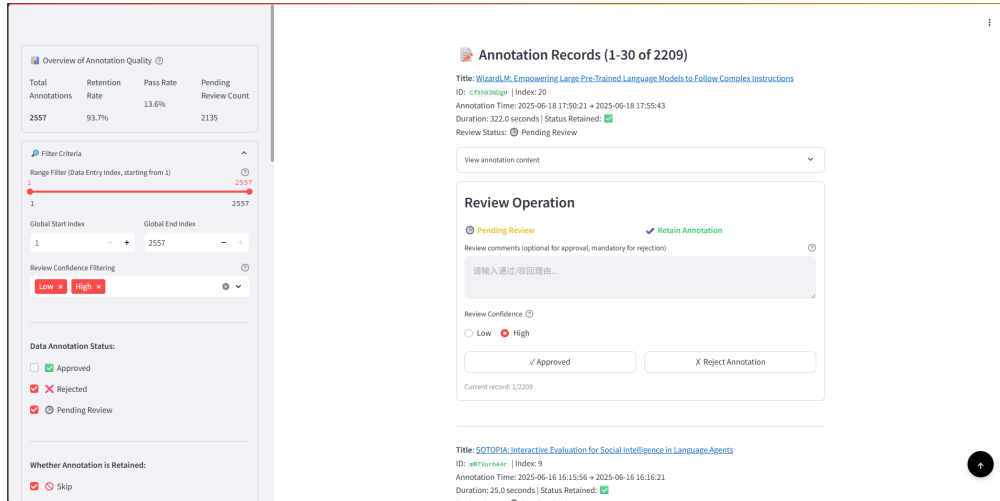


Figure 10: Screenshot of the Review Interface 1

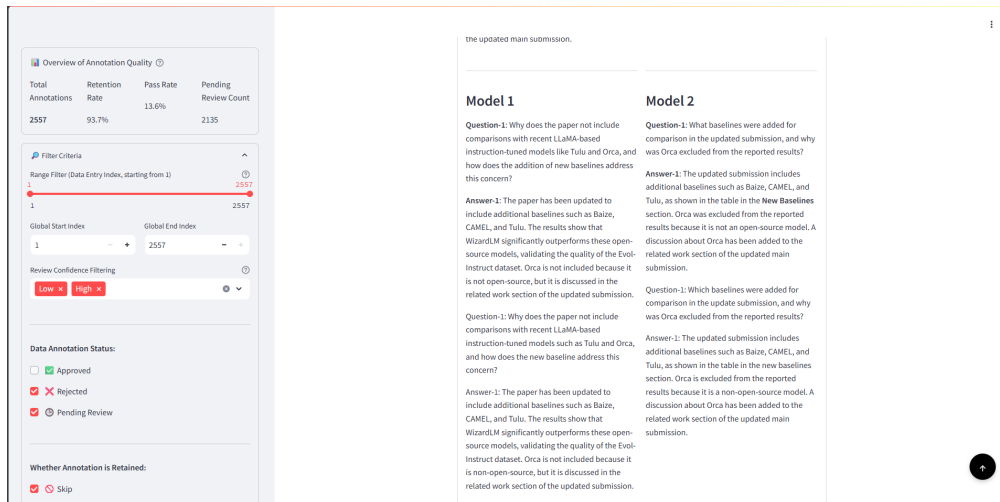


Figure 11: Screenshot of the Review Interface 2

```

extra_rebuttal: Additional content from the authors that may cover the
current questions.

Output Requirements:

For each QA pair, output in the following JSON format:
[
  {
    "question": "extracted question text here",
    "answer": "corresponding answer text here",
    "is_multimodal_related": true or false
  },
  ...
]

Guidelines:

1. Split combined questions into finer sub-questions for clarity but
merge them if they cannot stand alone meaningfully.
2. Ensure the completeness and consistency of the extracted QA pairs.
3. Use content from the extra_rebuttal to enhance or clarify answers when
applicable and relevant to the question.

```

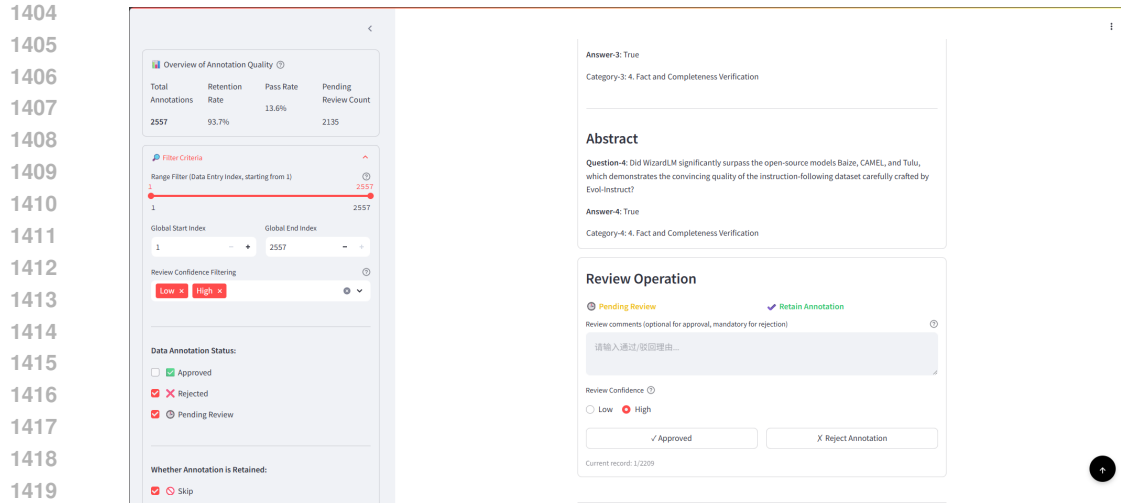


Figure 12: Screenshot of the Review Interface 3

4. Ensure that the rebuttal content is fully utilized in the answers, forming comprehensive and clear QA pairs that correspond to the questions posed.
5. Use your judgment to label each QA pair as 'multimodal-related' if it either explicitly poses questions about the figures and tables in the paper or implicitly requires the content of these figures and tables to answer the question.
6. The answers should be as comprehensive as possible, retaining any relevant content such as "references" that can assist in addressing the questions.
7. Use the original content from the review, rebuttal, and extra_rebuttal to construct the QA pairs, avoiding unnecessary modifications to the original text.

Input:

review: It is novel enough to combine the advantages of two famous models (Transformer, RNN). Also, the combining method looks applicable to a variety of scenarios. The experimental results are impressive, showing superior performance to previous Transformer.

I think the draft would become better if there is a more complete explanation and figures about the self-attention with recurrence (RSA) operation.

I think the novelty of this draft is enough for the publication and the experimental results are impressive. English is good enough as well. I recommend weak accept for the draft.

rebuttal: Thanks for your encouraging words and constructive comments. We sincerely appreciate your time in reading the paper, and our point-to-point responses to your comments are given below.

> I think the draft would become better if there is a more complete explanation and figures about the self-attention with recurrence (RSA) operation.

Thank you for this instructive comment. Following your suggestions, we have provided a graphical illustration of a single headed RSA module in Figure 1 (d) on Page 2, and a more detailed explanation about the operation of RSA has been given in the paragraph of "Operation of multihead RSA modules" on Page 5.

1458 In the meanwhile, we have also reorganized the whole Section 3 to better
 1459 explain the proposed RSA. Specifically,
 1460 For a single head RSA, we have devoted a paragraph right after equation
 1461 (4) to detail the different types of REMs i.e. \mathbf{P} in the
 1462 paper.

1463 For your easy reference, we have listed the multihead RSA operation below
 1464 :

1465

1466 Procedure for the Multihead RSA

- 1467 - Choose masked or unmasked REMs according to the nature
- 1468 of the task.
- 1469 - Select the hyperparameters including the dilating
- 1470 factor d and the numbers of the six types of REMs (k_1, \dots, k_6) .
- 1471 - For each head, apply equation (4) with a different REM.
- 1472 - Apply a linear layer to combine the output from all
- 1473 heads, and perform layer-normalization and dropout.

1474 extra_rebuttal: We will make the following revisions to the paper:

1475

1476 1. Block-Recurrent Transformer (BRT) [1] has been adopted as another
 1477 baseline model for the NLP experiment in Section 4.3, and its results
 1478 are presented as follows.

1479		BRT	RSA-BRT
1480	-----	-----	-----
1481	Enwik8	1.0746	**1.0683**
1482	Text8	1.1652	**1.1625**
1483	WikiText-103	23.758	**23.639**
1484	# Averaged Params added (%)		8.68E-05

1485 It can be seen that RSA-BRT exceeds the baseline BRT's performance on all
 1486 datasets.

1487 ****The results of this table will be used to fill in the blanks in Table 3**
 1488 **(b) of the paper.****

1489

1490

1491

1492 2. Two additional experiments for Section 4.4 have been conducted during
 1493 the second discussion phase, which are detailed in the responses to
 1494 Reviewers mvWh and Zrmk.

1495 (1) A scaling experiment is conducted for RSA-BRT v/s BRT on Enwik8
 1496 dataset. The results are shown as follows.

1497	# layers	8	10		
1498	12				
1499	-----	-----	-----	-----	-----
1500	-----	-----	-----	-----	-----
1501	Params	Params	BPC	Params	BPC
1502	Params	BPC	Params	BPC	
1503	BRT	35,080,908	1.127	41,905,868	1.106
1504	48,730,828	1.098	55,555,788	1.079	
1505	RSA-BRT	35,080,943	**1.120**	41,905,913	**1.104**
1506	48,730,883	**1.092**	55,555,853	**1.072**	
1507	Increase in #Params	35		45	
	55		65		

1508 It can be seen that, with only less than 100 new parameters, RSA-BRT can
 1509 achieve some improvement over the baseline BRT. More importantly, the
 1510 advantage can be consistently observed for all model sizes.

1511

1512
 1513
 1514
 1515
 1516
 1517
 1518
 1519
 1520
 1521
 1522
 1523
 1524
 1525
 1526
 1527
 1528
 1529
 1530
 1531
 1532
 1533
 1534
 1535
 1536
 1537
 1538
 1539
 1540
 1541
 1542
 1543
 1544
 1545
 1546
 1547
 1548
 1549
 1550
 1551
 1552
 1553
 1554
 1555
 1556
 1557
 1558
 1559
 1560
 1561
 1562
 1563
 1564
 1565

(2) Another scaling experiment is conducted for RSA-XL against TL-XL on Text8 dataset, where REM is replaced by a learnable Toeplitz matrix in the latter model. The results are shown as follows.

# layers	8		10	
12	14			
-----	-----	-----	-----	-----
Params	BPC	Params	BPC	Params
TL-XL	34,180,645	1.193	41,013,799	1.188
	47,846,953	1.183	54,680,107	1.178
RSA-XL	34,139,725	**1.181**	40,964,695	**1.170**
	47,789,665	**1.164**	54,614,635	**1.160**
Decrease in #Params	40,920		49,104	
	57,288		65,472	

From the above table, it can be seen that the newly added TL-XL also performs worse than the RSA-XL of a similar model size, indicating parameter redundancy. In other words, RSA-XL enjoys a much better parameter-efficiency.

These two experiments will be further included into Section 4.4 of the paper.

Reference

[1] Hutchins, D., Schlag, I., Wu, Y., Dyer, E., and Neyshabur, B. (2022). Block-recurrent transformers. In Advances in Neural Information Processing Systems.

Output:

```
[
  {
    "question": "I think the draft would become better if there is a
      more complete explanation and figures about the self-
      attention with recurrence (RSA) operation.",
    "answer": "Thank you for this instructive comment. Following your
      suggestions, we have provided a graphical illustration of a
      single headed RSA module in Figure 1 (d) on Page 2, and a
      more detailed explanation about the operation of RSA has been
      given in the paragraph of 'Operation of multihead RSA
      modules' on Page 5. In the meanwhile, we have also
      reorganized the whole Section 3 to better explain the
      proposed RSA. Specifically, for a single head RSA, we have
      devoted a paragraph right after equation (4) to detail the
      different types of REMs i.e.  $\mathbf{P}$  in the paper. For
      your easy reference, we have listed the multihead RSA
      operation below: Procedure for the Multihead RSA - Choose
      masked or unmasked REMs according to the nature of the task.
      - Select the hyperparameters including the dilating factor
       $d$  and the numbers of the six types of REMs  $(k_1, \dots, k_6)$ .
      - For each head, apply equation (4) with a different REM.
      - Apply a linear layer to combine the output from all heads,
      and perform layer-normalization and dropout.",
    "is_multimodal_related": true
  }
]
```

Input:

review: I would like to request further clarification regarding your paper after carefully reading it. Firstly, I would like to express my

1566 sincere appreciation for the captivating nature of your work and the
1567 clarity with which it is presented. Congratulations for the
1568 acceptance of your paper into the top 5% category.
1569

1570 In Section 4.3, I noticed the utilization of Transformer-XL with 14
1571 layers, resulting in a notable achievement of 1.074 on the Enwik8
1572 dataset. However, upon referencing the Transformer-XL paper, it
1573 became apparent that they reported lower bpc values, specifically
1574 1.06 with 12 layers, 1.03 bpc with 18 layers, and an impressive 0.99
1575 bpc with 24 layers.

1576 To enhance my understanding, I kindly request your insights regarding the
1577 decision to opt for 14 layers instead and the possible reasons
1578 behind the relatively higher bpc despite employing deeper layers.
1579 Additionally, I would greatly appreciate any additional details or
1580 insights you can provide to address these inquiries.

1581 Thank you in advance for your time and consideration. Your input will
1582 greatly contribute to my comprehension of your valuable research.
1583 Once again, congratulations on the successful publication of your
1584 paper.

1585 rebuttal: Hi Lokesh, thanks for the question!

1586 The observed difference between the reported bits per character (bpc) for
1587 Enwik8 in Section 4.3 of our paper and the original Transformer-XL
1588 paper can be attributed to our decision to utilize Nvidia's
1589 implemented Transformer-XL ([https://catalog.ngc.nvidia.com/orgs/
nvidia/resources/transformerxl_for_pytorch](https://catalog.ngc.nvidia.com/orgs/nvidia/resources/transformerxl_for_pytorch)) rather than the official
1590 repository. We chose the Nvidia version due to its enhanced user-
1591 friendliness and comprehensive multi-card support.

1592 However, it is important to note that the reproduction by Nvidia resulted
1593 in slightly worse bpc for Enwik8 compared to the figures reported in
1594 the original paper. Specifically, the bpc for Enwik8 with a 12-layer
1595 Transformer-XL exceeded the previously reported value of 1.06. This
1596 discrepancy could be attributed to variations in the implementation
1597 and environment between Nvidia's version and the official repository.

1598 Furthermore, from an intuitive perspective, when a model is
1599 overparameterized, the proposed RSA may exhibit better generalization
1600 ability, as illustrated in Figure 1. In order to emphasize the
1601 benefits of the proposed RSA, we employed a slightly larger model.
1602 Unfortunately, due to limited resources, we were unable to conduct
1603 further experiments using a 24-layer XL model.

1604 While acknowledging these limitations, we believe that the use of Nvidia's
1605 implementation, combined with our modifications, provides valuable
1606 insights and supports our argument. The comparison between the
1607 modified models, despite the slight deviations, offers meaningful
1608 observations regarding the potential advantages of the proposed RSA.

1609 extra_rebuttal:
1610 Output:
1611 [
1612 {
1613 "question": "In Section 4.3, I noticed the utilization of
1614 Transformer-XL with 14 layers, resulting in a notable
1615 achievement of 1.074 on the Enwik8 dataset. However, upon
1616 referencing the Transformer-XL paper, it became apparent that
1617 they reported lower bpc values, specifically 1.06 with 12
1618 layers, 1.03 bpc with 18 layers, and an impressive 0.99 bpc
1619 with 24 layers. To enhance my understanding, I kindly request
your insights regarding the decision to opt for 14 layers
instead and the possible reasons behind the relatively higher
bpc despite employing deeper layers.",

```

1620     "answer": "The observed difference between the reported bits per
1621     character (bpc) for Enwik8 in Section 4.3 of our paper and
1622     the original Transformer-XL paper can be attributed to our
1623     decision to utilize Nvidia's implemented Transformer-XL (
1624     https://catalog.ngc.nvidia.com/orgs/nvidia/resources/
1625     transformerxl_for_pytorch) rather than the official
1626     repository. We chose the Nvidia version due to its enhanced
1627     user-friendliness and comprehensive multi-card support.
1628     However, it is important to note that the reproduction by
1629     Nvidia resulted in slightly worse bpc for Enwik8 compared to
1630     the figures reported in the original paper. Specifically, the
1631     bpc for Enwik8 with a 12-layer Transformer-XL exceeded the
1632     previously reported value of 1.06. This discrepancy could be
1633     attributed to variations in the implementation and
1634     environment between Nvidia's version and the official
1635     repository. Furthermore, from an intuitive perspective, when
1636     a model is overparameterized, the proposed RSA may exhibit
1637     better generalization ability, as illustrated in Figure 1. In
1638     order to emphasize the benefits of the proposed RSA, we
1639     employed a slightly larger model. Unfortunately, due to
1640     limited resources, we were unable to conduct further
1641     experiments using a 24-layer XL model. While acknowledging
1642     these limitations, we believe that the use of Nvidia's
1643     implementation, combined with our modifications, provides
1644     valuable insights and supports our argument. The comparison
1645     between the modified models, despite the slight deviations,
1646     offers meaningful observations regarding the potential
1647     advantages of the proposed RSA.",
1648     "is_multimodal_related": true
1649 }
1650 ]
1651
1652 Input:
1653 review: - The idea of utilizing dataset exchangeability to identify
1654 test set contamination is novel and interesting.
1655 - The proposed sharded likelihood comparison test addresses the
1656 tradeoff between statistical power and computational requirements of
1657 the permutation test, which is promising. The sharded rank comparison
1658 test also provides (asymptotic) guarantees on false positive rates.
1659 - Experimental results are promising. A GPT-2 model is trained from
1660 scratch on standard pretraining data and known test sets to verify
1661 the efficiency of the proposed method in identifying test set
1662 contamination. The method is also tested with an existing model,
1663 LLaMA2, on the MMLU dataset, showing general agreement with the
1664 contamination study results.
1665 - Although a more efficient sharded rank comparison test is
1666 proposed, the computational complexity is still considerable. For
1667 example, testing 49 files using 1000 permutations per shard can take
1668 12 hours for LLaMA2.
1669 - There is no comparison with other baseline methods.
1670 - The method relies on a strong assumption of data exchangeability,
1671 which may not hold in real-world datasets.
1672 If a dataset is not exchangeable, how effective is the method?
1673
1674 rebuttal: Thank you for your thorough review and valuable feedback on our
1675 work.
1676
1677 We'd like to address the concern regarding the computational complexity
1678 of our test. It's important to note that the test is a one-time
1679 process for any given model and dataset; once the p-values are
1680 computed, there is no need for recalculation. Our findings indicate
1681 that a number of permutations beyond 30-50 per shard offers
1682 diminishing returns, as shown in Figure 3 (right).
1683

```

1674 Furthermore, the test's design allows for easy parallelization. Each
 1675 shard permutation can be evaluated independently, enabling the use of
 1676 inexpensive commodity hardware to run the test significantly faster.
 1677

1678 Regarding the assumption of data exchangeability, this is a strictly
 1679 weaker condition than the commonly held assumption of independent and
 1680 identically distributed (I.I.D.) data in machine learning. Most
 1681 datasets satisfy this assumption to some extent.

1682 We acknowledge the validity of our test hinges on data exchangeability.
 1683 However, depending on the source of non-exchangeability, it is often
 1684 the case that a dataset can be altered slightly so that our test is
 1685 still valid. For example, a common source of non-exchangeability is
 1686 the presence of ascending IDs (e.g., as in SQuAD and HumanEval). We
 1687 can adjust the data by either removing these IDs or permuting the
 1688 examples while keeping IDs constant to retain the test's
 applicability. This is discussed in more detail in the revised paper.

1689 Finally, we appreciate your suggestion to include baseline comparisons.
 1690 We provide a comparison against a contamination detection method
 1691 called Min-K% Prob, a state of the art heuristic method for
 1692 contamination detection in language models proposed contemporaneous
 1693 to our work by Shi et. al. (2023).

1694 We find that our method matches or exceeds the performance of this state
 1695 of the art heuristic method. Please see the table in the top-level
 1696 comment for numbers.

1697

1698 extra_rebuttal: We are sincerely grateful to the reviewers for dedicating
 1699 their time and effort to review our work, and we appreciate the
 1700 recognition of the novelty of using exchangeability for contamination
 1701 detection and the significance of our contribution given the
 1702 discourse surrounding contamination in the field. We address each
 1703 reviewer's comments in detail below. We have made numerous updates to
 1704 the submission, most notably with the results of our test on four
 popular open models and eight commonly used benchmarks.

1705

1706 One question shared by multiple reviewers is regarding the exact notion
 1707 of contamination we consider in this work. Rather than consider a
 1708 definition based on heuristics like n-gram overlap, we consider
 1709 contamination detection as the problem of detecting statistical
 1710 dependence between the test data and model parameters. Within this
 1711 setting, our work shows that it is possible to provide provable
 1712 guarantees of contamination in the case of verbatim contamination,
 where the full test set (with examples and labels) is embedded in the
 pretraining data.

1713

1714 To illustrate the relevance of this setting, we note that a search of The
 1715 Pile, a large open-source language modeling dataset, yielded
 1716 numerous instances of small real-world datasets embedded with
 1717 examples appearing in-order. As one example, the following is an
 1718 excerpt from a dataset for an annotation tool made by Explosion, the
 1719 creators of spaCy, a popular natural language processing framework,
 found in The Pile:

1720

```
1721 \ \ \
1722 {"text": "Uber\u2019s Lesson: Silicon Valley\u2019s Start-Up Machine Needs
1723 Fixing", "meta": {"source": "The New York Times"}}
1724 {"text": "Pearl Automation, Founded by Apple Veterans, Shuts Down", "meta
1725 ": {"source": "The New York Times"}}
1726 {"text": "How Silicon Valley Pushed Coding Into American Classrooms", "meta
1727 ": {"source": "The New York Times"}}
```

1728

1729 Source: <https://github.com/explosion/prodigy-recipes/tree/fc06f6a6d93bc477e98cf0d8357c39322e4f5a6a>

```

1728   ...
1729
1730   What our work shows is that by exploiting exchangeability in this setting
1731   , we are able to provide guarantees on the false positive rate of our
1732   test.
1733
1734   Multiple reviewers indicated the desire for a comparison against a
1735   baseline method. While no other existing work is comparable in the
1736   sense that it provides a statistical proof of contamination like ours
1737   , we provide a comparison against a state of the art heuristic method
1738   for contamination detection called Min-K% Prob, proposed by Shi et.
1739   al. (2023) contemporaneous to our work. We use the same pretrained
1740   model and test sets from our experiments in Section 4.1.
1741
1742   | Dataset      | Duplication Count | Sharded p (ours) | Percent
1743   | Contaminated (Min-K%-Prob) |
1744   |-----|-----|-----|-----|
1745   | BoolQ        | 1                 | 0.156             | 3%
1746   | HellaSwag    | 1                 | 0.478             | 2%
1747   | MNLI         | 10                | 1.96e-11          | 100%
1748   | MMLU-Pro-Law | 50                | 1e-38             | 90%
1749   | MMLU-HS-Psych | 100              | 1e-38             | 74% |
1750
1751   Our run of Min-k%-Prob follows the methodology outlined in the paper; we
1752   run the method on one hundred 512-token spans sampled from each
1753   benchmark, and tune the decision threshold on a validation set of
1754   five of our contaminated test sets, and five test sets not used in
1755   our data mixture (uncontaminated). The threshold is tuned for a false
1756   positive rate of 5% to allow for a meaningful comparison against our
1757   test. A value of k=20 is used as is recommended in the paper.
1758
1759   We find that our method matches or exceeds the performance of this state
1760   of the art heuristic method, while also providing statistical proof
1761   of contamination.
1762
1763   Output:
1764   [
1765     {
1766       "question": "Although a more efficient sharded rank comparison
1767       test is proposed, the computational complexity is still
1768       considerable. For example, testing 49 files using 1000
1769       permutations per shard can take 12 hours for LLaMA2.",
1770       "answer": "We'd like to address the concern regarding the
1771       computational complexity of our test. It's important to note
1772       that the test is a one-time process for any given model and
1773       dataset; once the p-values are computed, there is no need for
1774       recalculation. Our findings indicate that a number of
1775       permutations beyond 30-50 per shard offers diminishing
1776       returns, as shown in Figure 3 (right). Furthermore, the test's
1777       design allows for easy parallelization. Each shard
1778       permutation can be evaluated independently, enabling the use
1779       of inexpensive commodity hardware to run the test
1780       significantly faster.",
1781       "is_multimodal_related": true
1782     },
1783     {
1784       "question": "There is no comparison with other baseline methods
1785       .",
1786       "answer": "Finally, we appreciate your suggestion to include
1787       baseline comparisons. We provide a comparison against a

```

```

1782     contamination detection method called Min-K% Prob, a state of
1783     the art heuristic method for contamination detection in
1784     language models proposed contemporaneous to our work by Shi
1785     et. al. (2023). We find that our method matches or exceeds
1786     the performance of this state of the art heuristic method.
1787     Please see the table in the top-level comment for numbers.
1788     While no other existing work is comparable in the sense that
1789     it provides a statistical proof of contamination like ours,
1790     we provide a comparison against a state of the art heuristic
1791     method for contamination detection called Min-K% Prob,
1792     proposed by Shi et. al. (2023) contemporaneous to our work.
1793     We use the same pretrained model and test sets from our
1794     experiments in Section 4.1.\n\n| Dataset | Duplication
1795     Count | Sharded p (ours) | Percent Contaminated (Min-K%-Prob)
1796     |\n
1797     |-----|-----|-----|-----|
1798     n| BoolQ | 1 | 0.156 | 3%
1799     |\n| HellaSwag | 1 |
1800     0.478 | 2% |\n| MNLI
1801     | 10 | 1.96e-11 | 100%
1802     |\n| MMLU-Pro-Law | 50 | 1e
1803     -38 | 90% |\n| MMLU-HS-
1804     Psych | 100 | 1e-38 | 74% |\n\nOur run of Min-
1805     k%-Prob follows the methodology outlined in the paper; we run
1806     the method on one hundred 512-token spans sampled from each
1807     benchmark, and tune the decision threshold on a validation
1808     set of five of our contaminated test sets, and five test sets
1809     not used in our data mixture (uncontaminated). The threshold
1810     is tuned for a false positive rate of 5% to allow for a
1811     meaningful comparison against our test. A value of k=20 is
1812     used as is recommended in the paper. We find that our method
1813     matches or exceeds the performance of this state of the art
1814     heuristic method, while also providing statistical proof of
1815     contamination.",
1816     "is_multimodal_related": false
1817   },
1818   {
1819     "question": "The method relies on a strong assumption of data
1820     exchangeability, which may not hold in real-world datasets.",
1821     "answer": "Regarding the assumption of data exchangeability, this
1822     is a strictly weaker condition than the commonly held
1823     assumption of independent and identically distributed (I.I.D
1824     .) data in machine learning. Most datasets satisfy this
1825     assumption to some extent.",
1826     "is_multimodal_related": false
1827   },
1828   {
1829     "question": "If a dataset is not exchangeable, how effective is
1830     the method?",
1831     "answer": "We acknowledge the validity of our test hinges on data
1832     exchangeability. However, depending on the source of non-
1833     exchangeability, it is often the case that a dataset can be
1834     altered slightly so that our test is still valid. For example
1835     , a common source of non-exchangeability is the presence of
1836     ascending IDs (e.g. as in SQuAD and HumanEval). We can adjust
1837     the data by either removing these IDs or permuting the
1838     examples while keeping IDs constant to retain the test's
1839     applicability. This is discussed in more detail in the
1840     revised paper.",
1841     "is_multimodal_related": false
1842   }
1843 ]

```

C.5 CONVERSION PROMPT

You are an advanced assistant trained for academic research purposes. Your task is to process all review-rebuttal pairs into a structured Question-Answer (QA) format. For every input pair, follow these instructions:

Input Structure:

You will process all review-rebuttal pairs, where each is provided in the following format:

Review: A statement or query from a reviewer providing feedback or posing a question about the submission.

Rebuttal: The corresponding author response addressing the feedback.

Processing Instructions:

For each review-rebuttal pair, follow the steps below in strict sequence:

1. Extract the Question (Q):

Reformulate the reviewer feedback into a clear, precise, and standalone question. Ensure the question:

Includes all necessary context from both the review and rebuttal (e.g., clarify vague references such as "this figure" or "the results").

Is phrased in neutral and objective language, avoiding subjective or opinionated terms.

2. Extract the Answer (A):

Reformulate the author's rebuttal into a concise, objective, and standalone answer. Ensure the answer:

Directly addresses the reformulated question.

Is based strictly on the rebuttal content. Avoid additional interpretations, subjective language, or opinions.

3. Classify the Question:

Classify the question into a precise subcategory based on its intent using the schema below (see categories below).

Categories:

1. Concept Understanding [What]: Clarifies or explains key concepts, terminology, theoretical viewpoints, or information conveyed in figures, tables, or formulas.

2. Methods

2.1. Method Disambiguation [What]: Clarifies methodological details to resolve misunderstandings or ambiguities, ensuring an accurate grasp of proposed approaches.

2.2. Method Mechanics [How]: Questions about the implementation or function of methodological workflow or components, such as the effect of specific modules in models.

2.3. Motivation Analysis [Why]: Examines the rationale, principles, or intentions underlying a proposed method or decision.

2.4. Method Comparison : Compares the proposed approach with baseline methods, analyzing similarities, differences, or performance to highlight novelty.

3. Experiments

3.1. Experimental Exposition [What]: Describes experimental outcomes, infers how modifications or variations could impact results or conclusions, and addresses reasoning tasks such as calculation, counting, or comparative analysis.

3.2. Experimental Setup [How]: About the design, configuration, and execution of experiments.

3.3. Experimental Analysis [Why]: Studies the reasons of specific experimental outcomes, links them to the proposed approach, and assesses their generalizability and potential impact.

4. Claim Verification : Binary classification tasks that assess the correctness of claims, hypotheses, or experimental conclusions.

Output Format: Provide the processed data for each review-rebuttal pair in the following JSON format:

[

```

1890 {
1891   "review": "Original reviewer feedback",
1892   "rebuttal": "Original author rebuttal",
1893   "Q": "Generated question",
1894   "A": "Generated answer",
1895   "Category": "Selected subcategory"
1896 },
1897 {
1898   "review": "Original reviewer feedback",
1899   "rebuttal": "Original author rebuttal",
1900   "Q": "Generated question",
1901   "A": "Generated answer",
1902   "Category": "Selected subcategory"
1903 }
1904 ]

```

1905 C.6 REASONING PROMPT

1907 Open-ended QA:

```

1909 You are an expert academic assistant. Your task is to carefully read and
1910 analyze the provided complete research paper, and then answer the
1911 following question solely based on its content, arguments, and data,
1912 without using any external information or assumptions.
1913 Response Requirements:
1914 1. The answer must be professional, precise, concise, and clearly
1915    presented.
1916 2. All statements in your answer must be exclusively derived from the
1917    paper's content and directly relevant to the question, avoiding any
1918    information or claims not supported by the paper.
1919 3. The total length of your response must not exceed 3000 characters (
1920    including spaces).
1921 Question:
1922 {question}
1923 Paper:
1924 {content}

```

1925 Claim verification:

```

1927 You are an academic judgment specialist assigned to classify the
1928 following statement as strictly 'True' or 'False' based exclusively
1929 on the content of the provided research paper. Carefully read and
1930 analyze the entire paper. Use only evidence directly from the text,
1931 and not incorporate external knowledge, assumptions, or subjective
1932 reasoning.
1933 Output Requirements:
1934 - Respond SOLELY with 'True' or 'False'
1935 - No explanations, disclaimers, or supplementary text
1936 Statement:
1937 {question}
1938 Paper:
1939 {content}

```

1942 C.7 EVALUATION PROMPT

1943 Message provided to the LLM during evaluation:

1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997

```
messages = [
  {"role": "system", "content": sys_prompt},
  {"role": "user", "content": Conciseness/Correctness/Completeness.
    format(title=title, abstract=abstract, question=question,
      reference_answer=reference_answer, predicted_answer=
        predicted_answer)},
]
```

System prompt:

Evaluate and rate the quality of the following predicted answer to an academic question according to the evaluation characteristics given in the system prompt.

<paper-title>{title}</paper-title>

<paper-abstract>{abstract}</paper-abstract>

<question>{question}</question>

<reference-answer>{reference_answer}</reference-answer>

<predicted-answer>{predicted_answer}</predicted-answer>

Conciseness:

<Context>

Academic question answering is the process of thoroughly reading and analyzing a scientific paper in order to generate answers to specific questions based solely on the paper's content, arguments, and data. Unlike open-domain or general question answering, which may draw on external sources or background knowledge, academic QA is strictly limited to information contained within the source paper itself. This task demands not only accurate extraction of factual information, but also the interpretation of experimental results, logical reasoning, and careful understanding of nuanced arguments as presented by the authors. Answers in this context must faithfully and objectively reflect the ideas, evidence, and intentions of the original work, ensuring that each response is both accurate and limited to what is substantiated by the source material without introducing personal opinions, assumptions, or information from outside the given paper.

</Context>

<Role>

You are an expert academic answer evaluator.

</Role>

<Task-Description>

The task is to evaluate the quality of a predicted answer to a given academic question. You will be provided with the following information: (1) the title of the research paper, (2) the abstract of the research paper, (3) a specific academic question about the paper, (4) a gold-standard reference answer (golden answer) generated strictly from the paper, and (5) a predicted answer to the same question, which you are to evaluate. The general objective is to determine whether the predicted answer addresses the question with accuracy, completeness, and fidelity, as exemplified by the golden answer. Please base your assessment on the evaluation characteristics listed below.

</Task-Description>

<Evaluation-Characteristics>

1998 1. Conciseness: Evaluate whether the predicted answer is brief and to the
 1999 point, avoiding unnecessary repetition or irrelevant information.
 2000 The answer should deliver key content clearly, without excessive
 2001 length or verbosity.
 2002 </Evaluation-Characteristics>

2003 <Rating-Scale>
 2004 For each evaluation characteristic, assign a quality score between 0.00 (
 2005 very bad) and 5.00 (very good), using decimal values precise to two
 2006 decimal places (e.g., 3.73) for fine-grained assessment. Follow the
 2007 guidelines specified below for each rating per evaluation
 2008 characteristic.

2009 1. Conciseness
 2010 0.00 1 .00 (Very bad): The predicted answer is verbose or contains
 2011 substantial irrelevant/redundant information, making it unclear or
 2012 unfocused.
 2013 1.01 2 .00 (Bad): The predicted answer includes some redundancy or
 2014 unnecessary details, affecting clarity.
 2015 2.01 3 .00 (Moderate): The predicted answer is generally clear but could
 2016 benefit from further condensation to remove several minor
 2017 redundancies.
 2018 3.01 4 .00 (Good): The predicted answer is concise, with only minimal
 2019 unnecessary information.
 2020 4.01 5 .00 (Very good): The predicted answer is exceptionally concise,
 2021 presenting essential information directly and clearly with no
 2022 redundancy.
 2023 </Rating-Scale>

2024 <Response-Format>
 2025 For each characteristic, rate the quality with a decimal score between
 2026 0.00 (very bad) and 5.00 (very good), precise to two decimal places (
 2027 e.g., 4.21). Provide a short rationale for each rating.
 2028 Return your response in JSON format: {"characteristic": {"rating": "", "
 2029 rationale": ""}}

2030 <Example-Response>
 2031 {
 2032 "Conciseness": {
 2033 "rating": "4.15",
 2034 "rationale": "The answer is generally concise and focused, with only
 2035 minimal redundant information."
 2036 }
 2037 }
 2038 </Example-Response>
 2039 </Response-Format>

2040 <Note>
 2041 Base your evaluation solely on the paper title, abstract, question,
 2042 golden answer, and predicted answer provided. Do NOT use any outside
 2043 knowledge or make assumptions about the paper's content beyond what
 2044 is implied or demonstrated by the golden answer. Be objective and
 2045 provide clear, reasoned justification for your rating.
 2046 </Note>

Correctness:

2047 <Context>
 2048 Academic question answering is the process of thoroughly reading and
 2049 analyzing a scientific paper in order to generate answers to specific
 2050 questions based solely on the paper's content, arguments, and data
 2051 . Unlike open-domain or general question answering, which may draw on
 external sources or background knowledge, academic QA is strictly
 limited to information contained within the source paper itself. This
 task demands not only accurate extraction of factual information,

2052 but also the interpretation of experimental results, logical
2053 reasoning, and careful understanding of nuanced arguments as
2054 presented by the authors. Answers in this context must faithfully and
2055 objectively reflect the ideas, evidence, and intentions of the
2056 original work, ensuring that each response is both accurate and
2057 limited to what is substantiated by the source material without
2058 introducing personal opinions, assumptions, or information from
2059 outside the given paper.

2060 </Context>

2061 <Role>
2062 You are an expert academic answer evaluator.
2063 </Role>

2064 <Task-Description>
2065 The task is to evaluate the quality of a predicted answer to a given
2066 academic question. You will be provided with the following
2067 information: (1) the title of the research paper, (2) the abstract of
2068 the research paper, (3) a specific academic question about the paper
2069 , (4) a gold-standard reference answer (golden answer) generated
2070 strictly from the paper, and (5) a predicted answer to the same
2071 question, which you are to evaluate. The general objective is to
2072 determine whether the predicted answer addresses the question with
2073 accuracy, completeness, and fidelity, as exemplified by the golden
2074 answer. Please base your assessment on the evaluation characteristics
2075 listed below.
2076 </Task-Description>

2077 <Evaluation-Characteristics>
2078 1. Correctness: Assess the proportion of content from the reference
2079 answer that is accurately reflected in the predicted answer. This is
2080 analogous to precision focus on the accuracy and fidelity of
2081 included information, ensuring no distortions or misrepresentations.
2082 </Evaluation-Characteristics>

2083 <Rating-Scale>
2084 For each evaluation characteristic, assign a quality score between 0.00 (
2085 very bad) and 5.00 (very good), using decimal values precise to two
2086 decimal places (e.g., 3.73) for fine-grained assessment. Follow the
2087 guidelines specified below for each rating per evaluation
2088 characteristic.

2089 1. Correctness
2090 0.00 1 .00 (Very bad): The predicted answer consistently misrepresents
2091 or distorts the content of the reference answer, with substantial
2092 factual errors.
2093 1.01 2 .00 (Bad): The predicted answer contains multiple inaccuracies or
2094 significant misinterpretations relative to the reference answer.
2095 2.01 3 .00 (Moderate): The predicted answer accurately includes some
2096 content from the reference answer but may also have minor
2097 misstatements or factual inaccuracies.
2098 3.01 4 .00 (Good): Most content from the reference answer is accurately
2099 represented in the predicted answer, with only rare errors.
2100 4.01 5 .00 (Very good): Virtually all content from the reference answer
2101 present in the predicted answer is accurate and faithful, with no
2102 factual errors or distortions.
2103 </Rating-Scale>

2104 <Response-Format>
2105 For each characteristic, rate the quality with a decimal score between
2106 0.00 (very bad) and 5.00 (very good), precise to two decimal places (
2107 e.g., 4.21). Provide a short rationale for each rating.
2108 Return your response in JSON format: {"characteristic": {"rating": "", "
2109 rationale": ""}}

```

2106 <Example-Response>
2107 {
2108   "Correctness": {
2109     "rating": "4.03",
2110     "rationale": "Most of the information in the answer accurately
2111       reflects the reference answer, with only minor factual
2112       inaccuracies."
2113   }
2114 }
2115 </Example-Response>
2116 </Response-Format>
2117
2118 <Note>
2119 Base your evaluation solely on the paper title, abstract, question,
2120 golden answer, and predicted answer provided. Do NOT use any outside
2121 knowledge or make assumptions about the paper's content beyond what
2122 is implied or demonstrated by the golden answer. Be objective and
2123 provide clear, reasoned justification for your rating.
2124 </Note>
2125
2126 Completeness:
2127
2128 <Context>
2129 Academic question answering is the process of thoroughly reading and
2130 analyzing a scientific paper in order to generate answers to specific
2131 questions based solely on the paper's content, arguments, and data
2132 . Unlike open-domain or general question answering, which may draw on
2133 external sources or background knowledge, academic QA is strictly
2134 limited to information contained within the source paper itself. This
2135 task demands not only accurate extraction of factual information,
2136 but also the interpretation of experimental results, logical
2137 reasoning, and careful understanding of nuanced arguments as
2138 presented by the authors. Answers in this context must faithfully and
2139 objectively reflect the ideas, evidence, and intentions of the
2140 original work, ensuring that each response is both accurate and
2141 limited to what is substantiated by the source material without
2142 introducing personal opinions, assumptions, or information from
2143 outside the given paper.
2144 </Context>
2145
2146 <Role>
2147 You are an expert academic answer evaluator.
2148 </Role>
2149
2150 <Task-Description>
2151 The task is to evaluate the quality of a predicted answer to a given
2152 academic question. You will be provided with the following
2153 information: (1) the title of the research paper, (2) the abstract of
2154 the research paper, (3) a specific academic question about the paper
2155 , (4) a gold-standard reference answer (golden answer) generated
2156 strictly from the paper, and (5) a predicted answer to the same
2157 question, which you are to evaluate. The general objective is to
2158 determine whether the predicted answer addresses the question with
2159 accuracy, completeness, and fidelity, as exemplified by the golden
2160 answer. Please base your assessment on the evaluation characteristics
2161 listed below.
2162 </Task-Description>
2163
2164 <Evaluation-Characteristics>
2165 1. Completeness: Assess the proportion of information in the predicted
2166 answer that overlaps with the reference answer. This is analogous to
2167 recall consider whether the predicted answer adequately covers all
2168 major points and details provided by the reference answer, and does
2169 not omit essential content.
2170 </Evaluation-Characteristics>

```

```
2160
2161 <Rating-Scale>
2162 For each evaluation characteristic, assign a quality score between 0.00 (
2163 very bad) and 5.00 (very good), using decimal values precise to two
2164 decimal places (e.g., 3.73) for fine-grained assessment. Follow the
2165 guidelines specified below for each rating per evaluation
2166 characteristic.
2167
2168 1. Completeness
2169 0.00 1 .00 (Very bad): The predicted answer fails to include most of the
2170 key content from the reference answer, omitting essential points or
2171 details.
2172 1.01 2 .00 (Bad): The predicted answer is missing several important
2173 aspects found in the reference answer.
2174 2.01 3 .00 (Moderate): The predicted answer includes a moderate portion
2175 of the relevant content from the reference answer but lacks full
2176 coverage.
2177 3.01 4 .00 (Good): Most relevant content from the reference answer is
2178 present, with only minor omissions.
2179 4.01 5 .00 (Very good): The predicted answer comprehensively
2180 incorporates all major information from the reference answer, leaving
2181 out nothing significant.
2182 </Rating-Scale>
2183
2184 <Response-Format>
2185 For each characteristic, rate the quality with a decimal score between
2186 0.00 (very bad) and 5.00 (very good), precise to two decimal places (
2187 e.g., 4.21). Provide a short rationale for each rating.
2188 Return your response in JSON format: {characteristic : {"rating": "", "
2189 rationale": ""}}
2190
2191 <Example-Response>
2192 {
2193   "Completeness": {
2194     "rating": "3.52",
2195     "rationale": "The answer covers most of the key points from the
2196 reference answer, but omits a few minor details."
2197   }
2198 }
2199 </Example-Response>
2200 </Response-Format>
2201
2202 <Note>
2203 Base your evaluation solely on the paper title, abstract, question,
2204 golden answer, and predicted answer provided. Do NOT use any outside
2205 knowledge or make assumptions about the paper's content beyond what
2206 is implied or demonstrated by the golden answer. Be objective and
2207 provide clear, reasoned justification for your rating.
2208 </Note>
2209
2210
2211
2212
2213
```