AAAR-1.0: Assessing AI's Potential to Assist Research

Renze Lou[1], Hanzi Xu[2], Sijia Wang[3], Jiangshu Du[4], Ryo Kamoi[1], Xiaoxin Lu[1], Jian Xie[5], Yuxuan Sun[6], Yusen Zhang[1], Jihyun Janice Ahn[1], Hongchao Fang[1], Zhuoyang Zou[1], Wenchao Ma[1], Xi Li[7], Kai Zhang[8], Congying Xia[9], Lifu Huang[3], Wenpeng Yin[1]

[1]Pennsylvania State University; [2]Netflix; [3]University of California, Davis;
[4]University of Illinois Chicago; [5]Fudan University; [6]Zhejiang University;
[7]University of Alabama at Birmingham; [8]Ohio State University; [9]Salesforce Research
Correspondence to: {renze.lou, wenpeng}@psu.edu

Abstract

Numerous studies have assessed the proficiency of AI systems, particularly large language models (LLMs), in facilitating everyday tasks such as email writing, question answering, and creative content generation. However, researchers face unique challenges and opportunities in leveraging LLMs for their own work, such as brainstorming research ideas, designing experiments, and writing or reviewing papers. In this study, we introduce AAAR-1.0, a benchmark dataset designed to evaluate LLM performance in three fundamental, expertise-intensive research tasks: (i) EquationInference, assessing the correctness of equations based on the contextual information in paper submissions; (ii) ExperimentDesign, designing experiments to validate research ideas and solutions; and (iii) PaperWeakness, identifying weaknesses in paper submissions. AAAR-1.0 differs from prior benchmarks in two key ways: first, it is explicitly research-oriented, with tasks requiring deep domain expertise; second, it is researcher-oriented, mirroring the primary activities that researchers engage in on a daily basis. An evaluation of both open-source and proprietary LLMs reveals their potential as well as limitations in conducting sophisticated research tasks. We will release the AAAR-1.0 and keep iterating it to new versions.

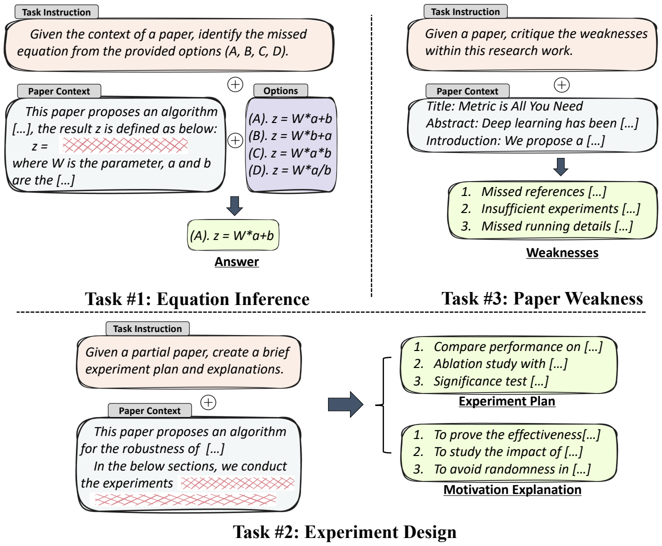

Task Instruction
Given the context of a paper, identify the missed equation from the provided options (A, B, C, D).
Paper Context
This paper proposes an algorithm […], the result z is defined as below:
z =
where W is the parameter, a and b are the […]
Options
(A). z = W*a+b
(B). z = W*b+a
(C). z = W*a*b
(D). z = W*a/b
(A). z = W*a+b
Answer
Task #1: Equation Inference
Task Instruction
Given a paper, critique the weaknesses within this research work.
Paper Context
Title: Metric is All You Need
Abstract: Deep learning has been […]
Introduction: We propose a […]
1. Missed references […]
2. Insufficient experiments […]
3. Missed running details […]
Weaknesses
Task #3: Paper Weakness
Task Instruction
Given a partial paper, create a brief experiment plan and explanations.
Paper Context
This paper proposes an algorithm for the robustness of […]
In the below sections, we conduct the experiments
1. Compare performance on […]
2. Ablation study with […]
3. Significance test […]
Experiment Plan
1. To prove the effectiveness[…]
2. To study the impact of […]
3. To avoid randomness in […]
Motivation Explanation
Task #2: Experiment Design

Figure 1: The input-output illustration of three tasks in the proposed AAAR-1.0 benchmark.

Introduction

Although AI has brought transformative changes to various aspects of life, its impact on researchers unfolds in a nuanced manner. On the one hand, AI assists in various research disciplines, such as Social Science, Finance, Medicine, GeoScience, Math, etc.(Yue et al. 2023; Li et al. 2023b), significantly expediting academic processes. However, many of these applications are superficial, often limited to data-driven clustering or classification. On the flip side, the AI era poses challenges for researchers. Despite its ability to streamline some activities, researchers still face demanding, cognitively intensive tasks such as staying current through extensive paper reading, rapidly generating ideas in response to fast-paced advancements, conducting rigorous experiments to substantiate claims, and managing an increasing volume of peer reviews. Then a question looms: How effectively can AI assist researchers in tasks that are domain-specific, expertise-demanding, and knowledge-intensive?

Existing works proved the promising potential for using LLMs in assisting AI research. Si, Yang, and Hashimoto (2024) conducted a large-scale human study and found that LLMs can generate creative research ideas. Lu et al. (2024) proposed an autonomous agent to handle complicated research workflow and write a whole research paper. However, most of these works focus on addressing highly subjective problems that require a high degree of expertise, making evaluation laborious and hard to reproduce. This underscores the need for a comprehensive benchmark that rigorously assesses LLMs' capabilities in expertise-intensive research activities

To this end, in this work, we introduce AAAR-1.0, a novel benchmark that aims to comprehensively assess the LLMs' capacity on expert-level research tasks. As illustrated in Figure 1, AAAR-1.0 decomposes three distinct expert-level AI research tasks from the researcher's daily activities, including i) EquationInference, investigating whether

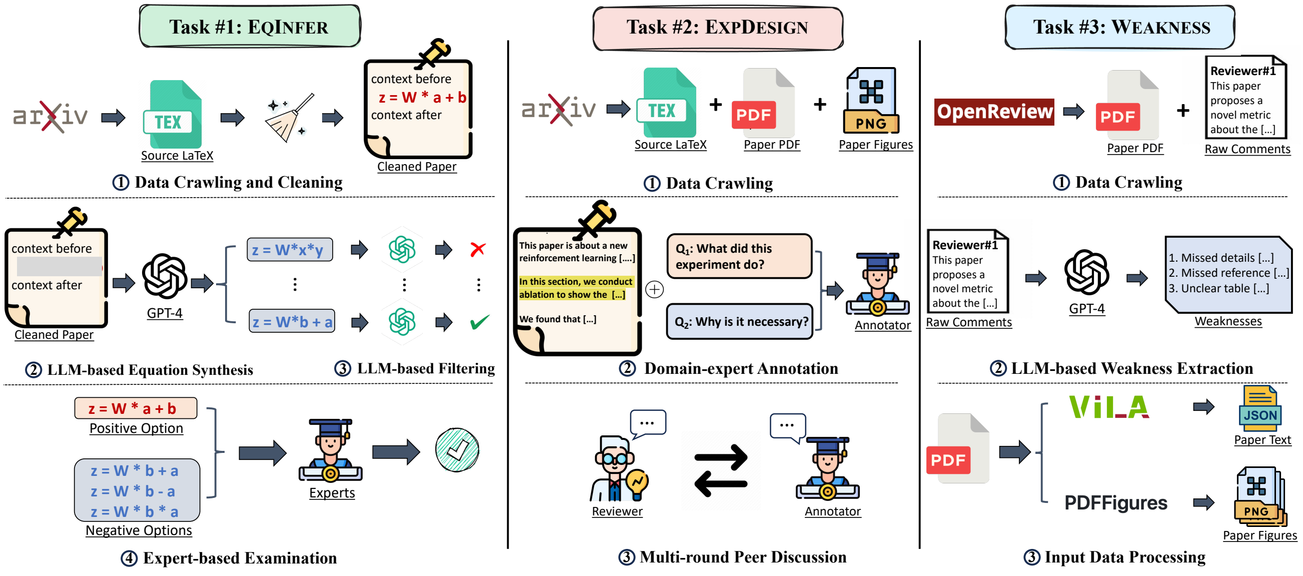


Figure 2: Data construction workflows of the three tasks in AAAR-1.0.

the LLMs can infer the equation correctness based on the paper context; ii) EXPERIMENTDESIGN, validating LLMs' ability on designing reliable experiments for a research idea; iii) PAPERWEAKNESS, testing the quality of the weaknesses criticism written by the LLMs. To ensure data quality, senior AI researchers with extensive domain expertise perform data annotation for AAAR-1.0, followed by rigorous multi-round data examination and filtering. All three tasks require models to possess strong domain knowledge covering various cutting-edge research findings, as well as expert-level research experience, to the extent that even humans need substantial research accumulation to tackle the tasks we designed. Crucially, tasks here are singular, stand-alone challenges (with clear input and output expectations) rather than a complicated task chain (Li et al. 2024; Lu et al. 2024), providing a more transparent assessment of the model's intermediate output.

Benefiting from the proposed automatic metrics, we conduct extensive experiments across numerous mainstream LLMs, where we find that:

- Closed-source LLMs generally outperform open-source LLMs on AAAR-1.0, likely due to their richer scientific knowledge stemming from a larger model size.
- Contrary to human behaviour, neither extending the input modality (i.e., leveraging text and figures) nor enlarging the input context guarantees enhanced performance. This underlines most current LLMs' limitations in processing diverse, extensive information coming from scientific documents.
- LLM-designed experiments are innovative and more diverse than those by humans; however, many are trivial, lack feasibility, and stray from the original research objectives.
- LLM-generated weaknesses often lack ample domain knowledge, especially on cutting-edge research topics, leading to the vague weaknesses applicable to various papers.

Related Work

LLMs for AI Research. With the rapid evolution of pertaining techniques, LLMs are found to be useful in assisting various research disciplines (Yu et al. 2024; Labrak et al. 2024), particularly in AI research, such as generating novel research ideas (Kumar et al. 2024), reviewing research draft (Gao, Brantley, and Joachims 2024; Du et al. 2024; Liang et al. 2024), and writing scientific papers (Chamoun, Schlichtkrull, and Vlachos 2024; Lu et al. 2024). For example, Si, Yang, and Hashimoto (2024) conducted a large-scale human investigation on LLM-generated research ideas and found that LLMs can generate novel ideas compared with humans while lacking feasibility. Du et al. (2024) found that while LLMs are effective at summarizing papers, they tend to overly trust the authors' claimed strengths and struggle to identify weaknesses specific to the paper. Furthermore, some works try to employ LLMs to solve more complicated research tasks that are composed of multiple steps (Li et al. 2024, 2023a; Tang et al. 2023). Notably, Lu et al. (2024) proposed AI-SCIENTIST, an autonomous agent framework that can handle a series of challenging research tasks consecutively, including generating research ideas, coming up with the corresponding experiments along with the implementations, and then writing the final research paper — exactly how human conduct a whole research pipeline. However, there is still a lack of systematic evaluations and quantitative analyses on the LLMs' (intermediate) output of each single-step research task. Our work focuses on building a benchmark that has individual research steps with clear input-output expectations, thus making it suitable for comprehensive LLMs evaluation.

Benchmarks for AI Research Tasks. Existing "LLM assists research" benchmarks mainly focus on the implementation and execution part of the research pipeline (Lu et al. 2024; Chen et al. 2024a; Li et al. 2024). For instance, Huang et al. (2024) proposed MLAgentBench to test the LLMs' capacity for writing project code and training the ML models, where the evaluation metric is the test performance of the models trained by LLMs. However, real-world AI research activities are diverse and some of them are hard to assess for quality, such as generating research ideas, which requires intensive manual assessment (Si, Yang, and Hashimoto 2024; Liang et al. 2024), or LLM-based estimation (Lu et al. 2024). Our work mainly focuses on high-level experience-based research tasks, and we try to build curated task-specific metrics for every single task for a more efficient and accurate LLMs appraisal.

AAAR-1.0

Figure 2 provides an overview of constructing AAAR-1.0. In the following sections, we elaborate on the data collection details of the aforementioned three tasks, including EQUATIONINFERENCE (**EQINFER**), EXPERIMENTDESIGN (**EXPDESIGN**), and PAPERWEAKNESS (**WEAKNESS**).

EQUATIONINFERENCE

Writing a correct scientific equation is challenging because it involves an in-depth understanding towards an algorithm or the relations between the massive notations. However, directly asking LLMs to generate equations is over-challenging. For this reason, in this task, we adopt the conventional multi-choice classification paradigm for building EQINFER , as shown in Figure 1.

① **Data crawling and cleaning.** For the data source, we adopt the pre-compilation LaTeX code for two reasons: i) existing PDF parsing tools, such as PyMuPDF and PaperMage (Lo et al. 2023), can introduce considerable noise to the parsed equation text; ii) considering most of exiting LLMs are capable with processing LaTeX code, using LaTeX source instead of parsed text can be more accurate and provide LLMs with richer information. Meanwhile, to avoid using any low-quality human-written equations, we only crawl those peer-reviewed papers accepted by top-tier conferences. Accordingly, we first obtain the accepted paper list from ACL Anthology, from year `2019` to `2023`. Next, we search each paper on arXiv to crawl its LaTeX source (if it exists). Finally, we get a total of 1,762 papers' source LaTeX packages.

We then clean the LaTeX sources by deleting all the comments and combining multiple cross-referred `.tex` files into a main file. Afterwards, we use regex to randomly extract (at most) 3 equations' code snippets per paper, finally resulting in 3,877 human-written equations are extracted.

② **LLM-based equation synthesis.** As we formulate this task as classification, for each human-written positive equation, we have to craft at least three counterpart negative equations. To this end, we prompt GPT-4 to synthesize more equations based on the paper context. For each positive equation, we repeat this prompting (with a high decoding temperature) until three different negative equations are synthesized.

③ **LLM-based filtering.** However, the LLM-synthetic equations can sometimes be context-unaligned, i.e., some synthesized equations contain notations that are never defined in the paper context, which is a superficial shortcut for the classification tasks (Geirhos et al. 2020). To improve the data quality, we prompt GPT-4 to identify those context-unaligned negative equations. We then discard those instances where all three negative equations are identified as contextually unaligned. This filtering leads to a final of 1,449 classification instances (62.3% instances are filtered).

④ **Expert-based examination.** Furthermore, it's also possible that synthesized negative equations are actually correct (i.e., false negative options) — even if the negative and positive equations are written differently, the final compiled results might be the same. To filter out the false negative equations and to have a final check on the classification instances, we then employ human experts to conduct a further data review.

We asked 5 senior PhD students who are experienced in AI research to manually check all the instances. For each classification instance, we ask human experts to consider the following criteria: i) **are all four equations (both positive and negative) grammatically correct?** ii) **after compilation, is there only one correct answer?** We ask every human expert to use external LaTeX compilation tools (e.g., TeXlive), and identify the instances that cannot meet the criteria. Each instance is examined by at least two experts, and we only keep instances that all experts decide to keep. After this strict examination, a total of 1,049 instances are eventually kept (27.6% instances are filtered)

Final data. We finally shuffle the four equations for each classification instance and randomly assign letters (`A`, `B`, `C`, and `D`) to the equations. We show the data statistics of the final EQINFER in Table 4 and the sample data cases in Appendix .

EXPERIMENTDESIGN

Given a research topic, such as a novel ML algorithm, a qualified researcher can design a solid experiment plan for it, and clarify underlying motivation to ensure the reliability of the designed experiment. Unlike the concurrent works that focus on the experiment implementation (Lu et al. 2024; Huang et al. 2024), we emphasize the importance of assessing the high-level experiment design of LLMs before the subsequent implementation to avoid any expensive execution iteration. Therefore, as shown in Figure 1, we formulate EXPDESIGN as a text-generation task that takes pre-experiment paper context as input, and then generates the experiment and explanation list.

① **Data crawling.** As for the data source, we first collect $\geq$ 10k papers' data from arXiv, including LaTeX sources and PDFs, which cover broad AI categories, including `cs.AI`, `cs.CL`, and `cs.CV`, from year `2018` to `2023`. Similarly, to ensure the source data quality, we only use papers that have appeared at well-known conferences.

② **Domain-expert annotation.** Making a reliable and executable experiment plan requires solid foundation knowledge of a specific research area. Consequently, we set a high standard for choosing annotators: i) be a senior PhD student with at least one peer-reviewed publication; ii) have more than 4 years of AI research experience; iii) frequently serve as conference reviewers. Finally, we invite a total of 10 qualified experts to participate in our data collection procedure. Given the 10k crawled papers, we first ask every annotator to bid on the papers that they are interested in. After bidding, each of them is assigned 10 papers by us, i.e., a total of 100 papers to be annotated. During annotation, we post each paper PDF on online Google Drive and ask the annotator to first carefully read the whole paper. Then, we ask them to identify and locate the key experiments in each paper (i.e., highlighting the relevant paragraphs of each experiment). We don't consider some trivial experiments, such as those supplemental analyses in the appendix section. For each identified experiment, the annotator has to concisely answer two questions: i) **what did this experiment do?** ii) **why did the paper authors conduct this experiment?** In other words, we ask the annotator to summarize all the key experiments in this paper and explain the underlying motivations based on their rich domain experience.

③ **Multi-round peer discussion.** Intuitively, different experts might have different opinions on the same research topic. Particularly, when explaining the underlying motivation of an experiment, adopting only a single expert's opinion might introduce bias to our annotation. Hence, we conduct a further multi-round peer discussion. For each online paper PDF, where all the key experiments are identified, summarized, and explained, we ask a different expert (reviewer) to review the annotation by considering the following three criteria: i) **are the identified experiments all the key experiments?** ii) **does each experiment summarization covers all key information?** iii) **does each explanation sound reasonable and reliable?** Each reviewer has to leave comments to the online PDF regarding the above criteria, and then the annotator has to respond to each comment — either accept the suggestion and revise the previous annotation, or provide a "rebuttal" to the reviewer to uphold the annotation. This discussion iterates until both opinions align with each other. Eventually, for each paper, we collect two lists: i) the experiment list, summarizing each experiment step of the paper; ii) the explanation list, the underlying motivations that are one-one corresponding to the experiment.

Final data. After annotation, we use the pre-experiment context of each paper (according to the first-experiment location identified by the annotator) as the input. Furthermore, we use GPT-4 to delete any sentence that potentially leaks the experiment from the input.[1] Similar to the EQINFER, we utilize the source LaTeX as the input text to avoid PDF paring noise. As for the image input, we collect those figures within each paper's source LaTeX package and only keep figures that are used in the pre-experiment context. Overall, a total of 100 instances are collected. As shown in Figure 1, the input of each instance is the pre-experiment context (including the figures), and the ground-truth output is the expert-annotated experiment plan and the explanations. Table 5 shows data statistics.

PAPERWEAKNESS

Another critical research task is paper review. Previous works have demonstrated the usefulness of the LLM-based review feedback (Gao, Brantley, and Joachims 2024; Jin et al. 2024; Lu et al. 2024). However, as indicated by Du et al. (2024); Liang et al. (2024), LLMs only excel at summarizing the research strengths while falling significantly short on weakness criticism. Hence, we build WEAKNESS for particularly investigating the LLM-generated weaknesses.

① **Data crawling.** We first crawl a total of 3,779 anonymous submissions of *ICLR 2023* from OpenReview,[2] including PDF and other meta information (e.g., scores, decisions, and tracks). As the *ICLR 2023* has 13 distinct tracks while the paper distribution across different tracks is highly biased, we then uniformly sample papers from different research tracks to improve the domain diversity. Meanwhile, during sampling, we also keep the accept/reject papers distributed equally to avoid data bias. In a word, we finally collect a total of 1,000 papers (500 accepted; 500 rejected), uniformly covering all 13 tracks. Please refer to Figure 3 for the track and score distribution of the 1,000 papers.

② **LLM-based weakness extraction.** Since the raw comments crawled from *ICLR 2023* are mixed with both strengths and weaknesses, we further employ GPT-4 to extract all the weaknesses from each reviewer's comments and compose multiple weaknesses into a list. Notably, we force GPT-4 to keep the original text of the reviewer, i.e., all weaknesses in our dataset are those original sentences written by the reviewer without any modifications.[3] What's more, sometimes one reviewer might repeatedly mention the same weakness throughout the comment. In this case, we simply keep all the repeated weaknesses because, if one weakness is repeatedly mentioned by the reviewer, it's intuitively an important weakness that the reviewer wants to emphasise; accordingly, keeping the repeat items can penalize LLMs more on missing this weakness.

All in all, for each paper, we can finally get multiple weakness lists (one weakness list per reviewer, one paper can have multiple reviewers). We further delete a few papers without any weaknesses found in the raw comments, resulting in a total of 993 instances, i.e., 993 {paper, weakness lists} pairs.

③ **Input data processing.** As we mentioned before, we crawl papers from OpenReview instead of arXiv because the under-review paper draft is required for this task. However, not every paper from OpenReview can be found on arXiv, i.e., the source LaTeX code and figures of most under-review papers are unavailable. Therefore, we utilize VILA (Lin et al.

[1] About 9.8% sentences are deleted.

[2] We adopt ICLR because it releases full submissions, while some other conferences only release accepted papers.

[3] We manually checked GPT-4's extraction results of 200 cases — GPT-4 only missed $\leq 1\%$ of reviewer-written weaknesses and maintained almost all the original text.

2023) to parse text data out from the PDF; we also employ PDFFigures-2.0 (Clark and Divvala 2016) to extract all the figures and tables (in image) from the paper, as Vila is not good at processing the table data.

Final data. Our final data is composed of 993 instances, each input is paper text along with figure/table images, and each output is peer reviewers' weakness lists. Table 6 shows data statistics.

Evaluation Criteria

For EqInfer, we adopt accuracy as the classification criterion. For ExpDesign and Weakness, since both tasks have natural language outputs, semantic-based metrics are necessary. Hence, in addition to the conventional ROUGE (Lin 2004), we also develop several novel similarity-based metrics for each specific task, including:

- **S-F$_1$** (equation 1 and 2): similarity-based F$_1$ for assessing the experiment design quality. It measures how well each model-generated experiment aligns with the human experiments.
- **S-Match** (equation 3): "soft" match score for evaluating the explanation. It calculates the similarity between human and model-generated explanations.
- **SN-F$_1$** (equation 4 and 5): updated version of S-F$_1$ to deal with the "nested" review weaknesses.
- **ITF-IDF** (equation 6): inspired by the classic TF-IDF; measures the inter- and intra-paper diversity of model-generated weaknesses.

We sincerely recommend referring to Appendix for the formal equation definitions of the above metrics.

Experiments and Analyses

In this section, we conduct extensive experiments on AAAR-1.0, across various popular LLMs, to quantify the current LLMs' capacity to tackle high-level research tasks. Specifically, the following sections include **EqInfer**, **ExpDesign**, and **Weakness**. Please refer to the Appendix for details on how to reproduce our experiment results.

EquationInference

Settings. As different LLMs have distinct context windows, to ensure a fair comparison, we fix the maximum input length for all models. According to the data statistics of Table 4, we empirically use 1,000 words for both contexts before and after equations, i.e., 2,000 surrounded words.

Main results. Table 1 shows the main results. Firstly, the open-source LLMs, especially the Falcon and Gemma, perform unexpectedly disappointing (even worse than random guesses). These screwed scores are mainly due to the poor long-context instruction following ability, where we find some open-source LLMs are confused with the massive input and often copy the LaTeX code from the input. In contrast, closed-source LLMs generally achieve superior accuracy, probably owing to the richer scientific knowledge from the larger model parameters. However, considering the conventional multi-choice QA formulation of EqInfer, the recently-released GPT-4o solely gets 43.18, implying the unique challenge of EqInfer compared with other scientific QA benchmarks (Song et al. 2023). Notably, with the help of internal CoT, o1 gains stronger performances than GPT-4/GPT-4o, indicating the potential benefits of adopting reasoning for this task.

Methods	**Accuracy (%)**
Random Guess	25.00
Open-source LLMs	
OLMo-7B (Groeneveld et al. 2024)	19.00
Falcon-40B (Almazrouei et al. 2023)	4.39
Gemma 2-27B (Gemma Team, 2024)	3.24
Mistral-7B (Jiang et al. 2023)	22.21
Mixtral-8x22B-MoE (Jiang et al. 2024)	37.08
Llama 3.1-70B (MetaAI 2024)	38.13
Qwen 2.5-72B (Qwen Team, 2024)	35.93
Closed-source LLMs	
Gemini 1.5 Pro (Anil et al. 2023)	34.31
Claude 3.5 sonnet (Anthropic 2024)	**61.10**
GPT-4 (OpenAI et al. 2023)	49.85
GPT-4o (OpenAI 2024a)	43.18
o1-preview (OpenAI 2024b)	59.49

Table 1: Various LLMs' performances on the 1,049 instances of EqInfer task.

$\mathcal{Q}$: do more contexts boost performance? Table 1 unifies the input context lengths to 1,000 words for various LLMs. In this paragraph, we experiment with long-context LLMs to investigate the impact of the input context lengths. Particularly, we scale the input length (per side) from 100 to 1,500 words. As shown in Figure 4, for the open-source LLMs (Llama and Qwen), after 300 words length, increasing the input context doesn't help the performance and even significantly drops Qwen's scores. While for the closed-source GPT-4-Turbo and GPT-4o, scaling up input length gradually boosts the performances at the first 1,000 words, but stabilizes afterwards. This is in line with human intuition, i.e., surrounding context is required for the equation inference, as the adjacent context usually provides important information, such as the target algorithm description or the notation definition. However, after exceeding a specific threshold, more context information is not beneficial anymore and even confuses those LLMs with poor long-context handling capacity (Wang et al. 2024; Liu et al. 2024).

ExperimentDesign

Settings. Similarly, we unify the input context length of different LLMs to ensure a fair comparison. According to Table 5, we set 2,000 and 3,000 input words for open- and closed-source LLMs, respectively. Meanwhile, as motivation explanation is the subsequent task of experiment design, using model-generated experiments can propagate errors in

Methods	Experiment Design			Motivation Explanation		
	S-F_1	S-Precision	S-Recall	S-Match	ROUGE-L	ROUGE-1
Copy Input	21.13	17.94	26.76	40.32	22.06	25.28
Open-source LLMs						
OLMo-7B (Groeneveld et al. 2024)	33.94	37.25	31.79	45.78	26.30	30.38
Falcon-40B (Almazrouei et al. 2023)	17.87	21.78	15.35	17.03	12.10	12.72
Gemma 2-27B (Gemma Team, 2024)	34.33	39.71	30.51	42.77	26.20	29.63
Mistral-7B (Jiang et al. 2023)	37.62	43.09	34.19	50.18	**30.20**	34.69
Mixtral-8x22B-MoE (Jiang et al. 2024)	42.21	50.13	36.82	49.07	29.96	34.53
Llama 3.1-70B (MetaAI 2024)	40.57	48.43	35.43	50.05	29.33	34.11
Qwen 2.5-72B (Qwen Team, 2024)	43.24	**51.73**	37.55	51.12	29.46	34.68
Closed-source LLMs						
Gemini 1.5 Pro (Anil et al. 2023)	51.87	50.77	53.37	52.87	28.52	33.80
Claude 3.5 sonnet (Anthropic 2024)	48.74	46.49	51.53	53.03	18.75	26.15
GPT-4 (OpenAI et al. 2023)	43.89	42.34	45.82	55.03	22.82	30.01
GPT-4o (OpenAI 2024a)	**53.00**	51.24	**55.12**	54.79	27.54	34.31
o1-preview (OpenAI 2024b)	46.67	45.04	48.70	**58.55**	29.11	**36.70**

Table 2: Various LLMs' performances on the 100 instances of ExpDesign . The motivation explanation is based on the oracle experiments to prevent error propagation. "Copy Input" is a random baseline: for experiment design, randomly select 5 sentences from the input paper; for motivation explanation, directly copy each experiment idea.

explanation, leading to inferior results for most LLMs. To this end, we provide LLMs with the oracle experiments when generating explanations.

Main results. Table 2 shows the main results. For the experiment design, the closed-source LLMs generally outperform open-source LLMs, and both closed-/open-source LLMs are superior to the "Copy Input" baseline (except the Falcon). Despite the higher S-Precision, the open-source LLMs are seriously deficient in S-Recall compared with closed-source LLMs ($\sim$10%↓). We find that closed-source LLMs are more creative in experiment design and tend to generate more experiment ideas than open-source LLMs (though most of the experiment ideas are trivial), leading to excellent S-Recall. As for the motivation explanation, the S-Match scores of closed-source LLMs still surpass the open-source LLMs, while the score difference is not significant. Furthermore, we find the negative correlation between S-Match and the ROUGE, where the ROUGE scores of closed-source LLMs are broadly inferior. We find that the open-source LLMs often try to copy the terms or phrases from the given experiment, or even simply paraphrase the experiment instead of explaining, which results in a high superficial overlap with the ground-truth explanation. This observation highlights the importance of adopting the proposed S-Match to avoid evaluation bias of traditional generation metrics.

$\mathcal{Q}_1$: can self-contained experiments enhance the explanation of motivation? When generating the explanation in Table 2, we provide LLMs with each individual experiment and let them explain one by one, because we find that, when providing the whole experiment list, those open-source models only explain partial experiments because of their poor instruction-following capacity. However, there are intuitively some semantic or logical relations between different experiments, e.g., some experiments are prerequisites to others. Therefore, this one-by-one prompting might break the self-containment of an experiment plan. Consequently, we test with the "whole-list" prompting, where the LLMs are given the complete experiment list and are asked to explain all experiment steps together.

As shown in Table 8, unlike the open-source LLMs, the explanation performances of those closed-source LLMs are generally improved after adopting whole-list prompting. According to further manual checking, after maintaining the self-containment of the experiments, the LLMs can refer to other experiments and better grasp the underlying motivation of the current experiment.

$\mathcal{Q}_2$: do human evaluation results align with automatic metrics for explanation? As the explanation can be open-ended, in this paragraph, we provide the human evaluation results on different LLMs' motivation explanation outputs. In detail, we randomly select 20 out of 100 papers and ask 5 annotators to read the experiments along with each model's explanations; we then let the annotator decide whether each model's explanation is acceptable (see Appendix for more details). Table 9 illustrates the results, where the score variance is higher than Table 2. However, the performance ranking of both tables is perfectly correlated with each other (Spearman's rank correlation coefficient = 1), demonstrating the effectiveness of S-Match.

$\mathcal{Q}_3$: do more contexts boost performance? We also investigate the impact of input context length for ExpDesign. As shown in Figure 5, we scale up the input pre-experiment context length from 0.1k to 10k words (10k words is the maximum paper context length in the dataset). For the experiment planning, more input context does improve the performance of different LLMs, while this benefit stops after exceeding 5k words, which is similar to EQINFER's scaling results — after the necessary information has been covered, scaling more up doesn't boost the performance. Meanwhile, the results of the motivation explanation demonstrate that explaining motivations almost doesn't require any paper context, i.e., the LLMs solely rely on the given experiments. However, we do not expect this because we hope LLMs can explain the motivation based on a thorough understanding of the paper, just like how human experts do. Hence, there is still a considerable gap between the LLMs and humans in terms of grasping research motivations.

$\mathcal{Q}_4$: does multi-modal input boost performance? Intuitively, besides the text, when designing experiments for a given research topic, the figures can provide rich supplementary information, such as an algorithm illustration that can help better understand this research topic and underlying motivations. Hence, we test different MLLMs' performances, including GPT4-o, GPT-4, and InternVL2 (Chen et al. 2024b). Table 10 shows the ablation results on the figure data. To our surprise, the figure data doesn't improve the MLLMs' results in this task, even harming the performances. This might be due to the low informativeness of the figures, as figures usually consume more input tokens but act only as supplementary information to the text, indicating future work on developing MLLMs that can effectively leverage the scientific figures.

PAPERWEAKNESS

Settings. Intuitively, the full paper context is necessary for conducting a review. Therefore, instead of setting a maximum input length, in WEAKNESS, we try to feed all the paper context into the LLMs. As the input length of WEAKNESS is extremely long (see Table 6), we adopt a "split-combine" method — we first split the whole paper into several smaller pieces and let LLMs predict the weaknesses of each piece separately; after that, we combine all pieces' weaknesses as a final complete prediction. In practice, for the length of each small piece, we set 2,000 and 3,000 words for open- and closed-source LLMs, respectively. Additionally, in this task, we also examine the performance of a recent agent framework, namely AI-SCI (Lu et al. 2024), which enhances GPT-4o's paper review ability by leveraging advanced prompting techniques, e.g., self-reflection (Shinn et al. 2024) and response ensembling (Wang et al. 2023).[4]

Main results. Table 3 shows the main results, where the closed-source LLMs' overall performances are generally superior to the results of open-source LLMs. Similarly, closed-source LLMs are particularly excellent in SN-Recall because of more generated weaknesses. However, there is still a considerable gap in the weakness diversity between the LLMs and human experts.[5] Compared with human review, most LLM-generated weaknesses are vague and lack the necessary knowledge about some frontier research works. Surprisingly, AI-SCI performs worse than backbone GPT-4o, especially on ITF-IDF, which suggests the challenge of WEAKNESS, i.e., simply adopting popular prompting techniques cannot well address this task.

$\mathcal{Q}_1$: is the split-combine effective? Ideally, if the LLM has a sufficient context window size, it is not that necessary to split the input papers for separate processing. Consequently, in this paragraph, we utilize the LLMs accepting long context input to compare "split-combine" with "no-split", i.e., letting LLMs write weaknesses by giving the full paper. In practice, we set the maximum number of input words to 20k, which ensures $\geq$95% papers in the WEAKNESS can be fully processed. As shown in Table 7, compared with giving the full paper contexts, split-combine generally brings about superior performances. During manual checking, we find that, when full paper is available, LLMs frequently neglect some important sections and omit weaknesses accordingly, while split-combine ensures that the LLMs can carefully brainstorm weaknesses within each smaller piece. Surprisingly, the LLMs' performances with full paper context can be even worse than just remaining the first 3,000 words. This implies that even the current powerful long-context LLMs still fall short when processing long scientific documents (Liu et al. 2024).

$\mathcal{Q}_2$: does multi-modal input boost performance? Our dataset covers both tables and figure illustrations extracted from the paper PDF as inputs. Intuitively, when reviewing a paper, both figures and tables are critical, not only for a better understanding, but also because some weaknesses are related to tables/figures.[6] Therefore, in Table 11, we adopt two MLLMs to investigate the effectiveness of image inputs. Overall, image information, including both figures and tables, doesn't bring significant performance improvement, i.e., only InternVL2 gains a performance boost after incorporating figures; while tables slightly drop both models' results. This is probably because the MLLMs cannot reason well over the information-intensive images, especially the table images (Deng et al. 2024).

Conclusion

In this work, we propose AAAR-1.0, a novel benchmark targeting a comprehensive evaluation of the current LLMs' AI research capacity. We devise three distinct expertise-intensive tasks along with the curated evaluation metrics, and collect high-quality data by employing senior AI researchers. Multi-round strict data examination and filtering are conducted to try our best to avoid any significant noise in the data. Extensive experiments across various mainstream LLMs highlight the challenges and values of AAAR-1.0, where there is still a considerable gap between LLMs and human experts.

[4] We don't run AI-SCI on EXPDESIGN, because AI-SCI takes model-generated ideas as the inputs, which are incompatible with our task setting.

[5] Note that the human's ITF-IDF score in Table 3 can be slightly underestimated. This is because we keep the repeated weaknesses in the human review, which affects the human review's informativeness (lower ITF) but is useful when calculating the SN-Recall for LLMs.

[6] We find that there is approximately one human-written weakness related to figures or tables in each paper.

Methods	SN-F$_1$ (%)	SN-Precision (%)	SN-Recall (%)	Review Diversity
				ITF-IDF (↑)
Peer Review	—	—	—	7.69
Open-source LLMs				
OLMo-7B (Groeneveld et al. 2024)	43.25	40.38	47.04	2.45
Falcon-40B (Almazrouei et al. 2023)	27.34	25.13	30.88	1.06
Gemma 2-27B (Gemma Team, 2024)	35.85	34.68	37.91	1.43
Mistral-7B (Jiang et al. 2023)	42.03	43.80	40.77	1.17
Mixtral-8x22B-MoE (Jiang et al. 2024)	43.23	**44.59**	42.23	0.98
Llama 3.1-70B (MetaAI 2024)	42.78	43.19	42.70	2.60
Qwen 2.5-72B (Qwen Team, 2024)	42.74	43.80	42.05	1.21
Closed-source LLMs				
Gemini 1.5 Pro (Anil et al. 2023)	**48.75**	43.97	55.08	5.88
Claude 3.5 sonnet (Anthropic 2024)	47.85	41.97	56.00	3.91
GPT-4 (OpenAI et al. 2023)	47.66	42.15	55.19	5.31
GPT-4o (OpenAI 2024a)	47.73	42.09	55.48	**5.95**
o1-preview (OpenAI 2024b)	48.62	42.54	**57.08**	5.63
LLM Agent Framework				
AI-SCI (GPT-4o) (Lu et al. 2024)	45.05	40.02	51.91	2.23

Table 3: Various LLMs' performances on the 993 instances of Weakness.

References

Almazrouei, E.; Alobeidli, H.; Alshamsi, A.; Cappelli, A.; Cojocaru, R.; Debbah, M.; Goffinet, E.; Heslow, D.; Launay, J.; Malartic, Q.; Noune, B.; Pannier, B.; and Penedo, G. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Anil, R.; Borgeaud, S.; Wu, Y.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; Team, G.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Anthropic. 2024. Introducing Claude 3.5 Sonnet. https://www.anthropic.com/news/claude-3-5-sonnet.

Chamoun, E.; Schlichktrull, M.; and Vlachos, A. 2024. Automated Focused Feedback Generation for Scientific Writing Assistance. *arXiv preprint arXiv:2405.20477*.

Chen, Z.; Chen, S.; Ning, Y.; Zhang, Q.; Wang, B.; Yu, B.; Li, Y.; Liao, Z.; Wei, C.; Lu, Z.; et al. 2024a. ScienceAgentBench: Toward Rigorous Assessment of Language Agents for Data-Driven Scientific Discovery. *arXiv preprint arXiv:2410.05080*.

Chen, Z.; Wang, W.; Tian, H.; Ye, S.; Gao, Z.; Cui, E.; Tong, W.; Hu, K.; Luo, J.; Ma, Z.; et al. 2024b. How Far Are We to GPT-4V? Closing the Gap to Commercial Multimodal Models with Open-Source Suites. *arXiv preprint arXiv:2404.16821*.

Clark, C.; and Divvala, S. 2016. Pdffigures 2.0: Mining figures from research papers. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*, 143–152.

Deng, N.; Sun, Z.; He, R.; Sikka, A.; Chen, Y.; Ma, L.; Zhang, Y.; and Mihalcea, R. 2024. Tables as Texts or Images: Evaluating the Table Reasoning Ability of LLMs and MLLMs. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics ACL 2024*, 407–426. Bangkok, Thailand and virtual meeting: Association for Computational Linguistics.

Du, J.; Wang, Y.; Zhao, W.; Deng, Z.; Liu, S.; Lou, R.; Zou, H. P.; Venkit, P. N.; Zhang, N.; Srinath, M.; Zhang, H. R.; Gupta, V.; Li, Y.; Li, T.; Wang, F.; Liu, Q.; Liu, T.; Gao, P.; Xia, C.; Xing, C.; Cheng, J.; Wang, Z.; Su, Y.; Shah, R. S.; Guo, R.; Gu, J.; Li, H.; Wei, K.; Wang, Z.; Cheng, L.; Ranathunga, S.; Fang, M.; Fu, J.; Liu, F.; Huang, R.; Blanco, E.; Cao, Y.; Zhang, R.; Yu, P. S.; and Yin, W. 2024. LLMs Assist NLP Researchers: Critique Paper (Meta-)Reviewing. In *The 2024 Conference on Empirical Methods in Natural Language Processing*.

Gao, Z.; Brantley, K.; and Joachims, T. 2024. Reviewer2: Optimizing Review Generation Through Prompt Generation. *arXiv preprint arXiv:2402.10886*.

Geirhos, R.; Jacobsen, J.-H.; Michaelis, C.; Zemel, R.; Brendel, W.; Bethge, M.; and Wichmann, F. A. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11): 665–673.

Groeneveld, D.; Beltagy, I.; Walsh, P.; Bhagia, A.; Kinney, R.; Tafjord, O.; Jha, A. H.; Ivison, H.; Magnusson, I.; Wang, Y.; Arora, S.; Atkinson, D.; Authur, R.; Chandu, K.; Cohan, A.; Dumas, J.; Elazar, Y.; Gu, Y.; Hessel, J.; Khot, T.; Merrill, W.; Morrison, J.; Muennighoff, N.; Naik, A.; Nam, C.; Peters, M. E.; Pyatkin, V.; Ravichander, A.; Schwenk, D.; Shah, S.; Smith, W.; Subramani, N.; Wortsman, M.; Dasigi, P.; Lambert, N.; Richardson, K.; Dodge, J.; Lo, K.; Soldaini, L.; Smith, N. A.; and Hajishirzi, H. 2024. OLMo: Accelerating the Science of Language Models. *Preprint*.

Huang, Q.; Vora, J.; Liang, P.; and Leskovec, J. 2024. MLAgentBench: Evaluating Language Agents on Machine Learning Experimentation. In *Forty-first International Conference on Machine Learning*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Jiang, A. Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Hanna, E. B.; Bressand, F.; et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Jin, Y.; Zhao, Q.; Wang, Y.; Chen, H.; Zhu, K.; Xiao, Y.; and Wang, J. 2024. AgentReview: Exploring Peer Review Dynamics with LLM Agents. *arXiv preprint arXiv:2406.12708*.

Kumar, S.; Ghosal, T.; Goyal, V.; and Ekbal, A. 2024. Can Large Language Models Unlock Novel Scientific Research Ideas? *arXiv preprint arXiv:2409.06185*.

Labrak, Y.; Bazoge, A.; Morin, E.; Gourraud, P.-A.; Rouvier, M.; and Dufour, R. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. *arXiv preprint arXiv:2402.10373*.

Li, H.; Jiang, H.; Zhang, T.; Yu, Z.; Yin, A.; Cheng, H.; Fu, S.; Zhang, Y.; and He, W. 2023a. TrainerAgent: Customizable and Efficient Model Training through LLM-Powered Multi-Agent System. *arXiv preprint arXiv:2311.06622*.

Li, R.; Patel, T.; Wang, Q.; and Du, X. 2024. MLR-Copilot: Autonomous Machine Learning Research based on Large Language Models Agents. *arXiv preprint arXiv:2408.14033*.

Li, Z.; Zhou, W.; Chiang, Y.-Y.; and Chen, M. 2023b. Geolm: Empowering language models for geospatially grounded language understanding. *arXiv preprint arXiv:2310.14478*.

Liang, W.; Zhang, Y.; Cao, H.; Wang, B.; Ding, D. Y.; Yang, X.; Vodrahalli, K.; He, S.; Smith, D. S.; Yin, Y.; et al. 2024. Can large language models provide useful feedback on research papers? A large-scale empirical analysis. *NEJM AI*, 1(8): AIoa2400196.

Lin, C.-Y. 2004. Rouge: A Package for Automatic Evaluation of Summaries. In *Text summarization branches out*, 74–81.

Lin, J.; Yin, H.; Ping, W.; Lu, Y.; Molchanov, P.; Tao, A.; Mao, H.; Kautz, J.; Shoeybi, M.; and Han, S. 2023. VILA: On Pre-training for Visual Language Models. arXiv:2312.07533.

Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12: 157–173.

Lo, K.; Shen, Z.; Newman, B.; Chang, J. Z.; Authur, R.; Bransom, E.; Candra, S.; Chandrasekhar, Y.; Huff, R.; Kuehl, B.; et al. 2023. PaperMage: A Unified Toolkit for Processing, Representing, and Manipulating Visually-Rich Scientific Documents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 495–507.

Lu, C.; Lu, C.; Lange, R. T.; Foerster, J.; Clune, J.; and Ha, D. 2024. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. *arXiv preprint arXiv:2408.06292*.

MetaAI. 2024. Introducing Llama 3.1: Our most capable models to date. https://ai.meta.com/blog/meta-llama-3-1/.

OpenAI. 2024a. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/.

OpenAI. 2024b. Introducing OpenAI o1. https://openai.com/index/introducing-openai-o1-preview/.

OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Reimers, N. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Si, C.; Yang, D.; and Hashimoto, T. 2024. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. *arXiv preprint arXiv:2409.04109*.

Song, L.; Zhang, J.; Cheng, L.; Zhou, P.; Zhou, T.; and Li, I. 2023. Nlpbench: Evaluating large language models on solving nlp problems. *arXiv preprint arXiv:2309.15630*.

Tang, X.; Liu, Y.; Cai, Z.; Shao, Y.; Lu, J.; Zhang, Y.; Deng, Z.; Hu, H.; An, K.; Huang, R.; et al. 2023. ML-Bench: Evaluating Large Language Models and Agents for Machine Learning Tasks on Repository-Level Code. *arXiv e-prints*, arXiv–2311.

Team, G. 2024a. Google launches Gemma 2, its next generation of open models. https://blog.google/technology/developers/google-gemma-2/.

Team, Q. 2024b. Qwen2.5: A Party of Foundation Models.

Wang, M.; Chen, L.; Fu, C.; Liao, S.; Zhang, X.; Wu, B.; Yu, H.; Xu, N.; Zhang, L.; Luo, R.; et al. 2024. Leave no document behind: Benchmarking long-context llms with extended multi-doc qa. *arXiv preprint arXiv:2406.17419*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yu, B.; Baker, F. N.; Chen, Z.; Ning, X.; and Sun, H. 2024. Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset. *arXiv preprint arXiv:2402.09391*.

Yue, X.; Ni, Y.; Zhang, K.; Zheng, T.; Liu, R.; Zhang, G.; Stevens, S.; Jiang, D.; Ren, W.; Sun, Y.; et al. 2023. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*.

Appendices

Within this supplementary material, we elaborate on the following aspects:

- Formal Definition of Evaluation Criteria
- Data Statistics and Diversity
- Implementation Details
- More Experiment Results
- Prompt Templates

Formal Definition of Evaluation Criteria

For the experiment plan list of ExpDesign, we hope the LLMs can mention as many similar experiment steps as the expert's plan. Nevertheless, we also don't expect LLMs to generate too many irrelevant or redundant steps in the plan. This intuition covers both the "recall" and "precision" aspects. Therefore, we develop semantic similarity-based F_1 score, denoted as S-F_1, which is the harmonic mean of S-Precision and S-Recall:

$$\text{S-Precision} = \frac{1}{m}\sum_{i=1}^{m}\max_{j}\text{sim}(p_i, g_j) \tag{1}$$

$$\text{S-Recall} = \frac{1}{n}\sum_{j=1}^{n}\max_{i}\text{sim}(g_j, p_i) \tag{2}$$

where the p and g represent the LLM's prediction plan and the ground-truth plan, respectively. The m and n are the list length of p and g (e.g., m experiment steps in p). We use SentenceBERT (Reimers 2019) to measure the semantic similarity between the p_i step and the g_j step.

Meanwhile, S-F_1 omits the item order difference of two lists, but when giving same-length lists (items have one-one correspondence), we can utilize the following similarity-based matching score:

$$\text{S-Match} = \frac{1}{m}\sum_{i=1}^{m}\text{sim}(p_i, g_i) \tag{3}$$

Unlike ExpDesign, the output of Weakness is multiple reviewers' weakness lists, which means we have to measure LLM's single prediction list with a "nested" list. Hence, we rewrite S-Precision , S-Recall to SN-Precision , SN-Recall:

$$\text{SN-Precision} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{r}\sum_{k=1}^{r}\max_{j}\text{sim}(p_i, g_j^k)\right) \tag{4}$$

$$\text{SN-Recall} = \frac{1}{r}\sum_{k=1}^{r}\left(\frac{1}{n_k}\sum_{j=1}^{n_k}\max_{i}\text{sim}(g_j^k, p_i)\right) \tag{5}$$

where r is the number of reviewers of the given paper, n_k means the length of k-th reviewer's weakness list, and g_j^k indicates the j-th item in k-th reviewer's weakness list.

Additionally, in the real world, we would think a review weakness is reliable if it is specific to a paper. Meanwhile, we also hope the review is informative, i.e., no excessive similar weaknesses in one review. Inspired by the classic TF-IDF, we propose a novel review diversity metric:

$$\text{ITF-IDF} = \frac{1}{w}\sum_{j=1}^{w}\left(\frac{1}{m_j}\sum_{i=1}^{m_j}\log\left(\frac{m_j}{O_i^j}\right)\times\log\left(\frac{w}{R_i^j}\right)\right) \tag{6}$$

$$O_i^j = \sum_{k=1}^{m_j}\text{sim}(p_i^j, p_k^j) \tag{7}$$

$$R_i^j = \sum_{l=1}^{w}\max_{s}\text{sim}(p_i^j, p_s^l) \tag{8}$$

where the w is the total number of papers in the dataset, p^j is j-th paper's prediction weakness list, p_i^j is the i-th weakness in p^j. Moreover, O_i^j calculates the intra-paper occurrence frequency of p_i^j; R_i^j is the "soft" number of papers that also contain the p_i^j, which is computed by summing the maximum similarity scores between p_i^j and other paper's weaknesses. In a word, O_i^j measures informativeness, and R_i^j measures specificity. The complete ITF-IDF consider both aspects and reflects the overall weakness diversity.

Data Statistics and Diversity

We provide the detailed data statistics of three datasets in our benchmark, as shown in Table 4, 5, and 6. We use the NLTK package[7] to tokenize words and count the length. When calculating the length of equations, we use the pylatexenc tool[8] to simplify the equations first.

Meanwhile, for the Weakness, we also plot the review scores distribution of the papers used in the dataset, as well as the track distribution. As can be found in Figure 3, our dataset has a decent distribution, where the papers are uniformly distributed across 13 tracks, and most papers' scores ranged from 5 to 8 (i.e., most papers are weakly rejected or accepted).

Implementation Details

Metric Details

When calculating the metrics, specifically for the similarity-based scores, we utilize SentenceBERT (Reimers 2019) to encode each segment (e.g., each experiment idea in the list) into a dense vector, and then calculate the cosine similarity,[9] which takes about 1GB of memory when running on a single A100 GPU.

LLMs Running Details

In our experiments, we utilize various LLMs, including both closed and open-sourced. We list the model weight sources for the open-source LLMs:

- OLMo-7B: https://huggingface.co/allenai/OLMo-7B
- Falcon-40B: https://huggingface.co/tiiuae/falcon-40b
- Gemma 2-27B: https://huggingface.co/google/gemma-2-27b

[7] https://www.nltk.org/

[8] https://github.com/phfaist/pylatexenc

[9] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

# of classification instances	1,049
# of source papers	869
ave. “`left`” input context length (in words)	4,377
ave. “`right`” input context length (in words)	6,362
max “`left`” input context length (in words)	24,849
max “`right`” input context length (in words)	32,948
min “`left`” input context length (in words)	711
min “`right`” input context length (in words)	8
ave. “`pos.`” output equation length (in character)	55
ave. “`neg.`” output equation length (in character)	48
max “`pos.`” output equation length (in character)	1,039
max “`neg.`” output equation length (in character)	306
min “`pos.`” output equation length (in character)	6
min “`neg.`” output equation length (in character)	4

Table 4: The statistics of EqInfer . Here, the “`left`” and “`right`” input context indicates the paper contexts before and after the missed equation; “`pos.`” means the ground-truth equations (written by the source paper authors), while “`neg.`” is the GPT4-synthetic wrong equations.

# of instances	100
# of source papers	100
ave. input context length (in words)	4,288
max input context length (in words)	9,799
min input context length (in words)	698
ave. # of input figures	2.6
max # of input figures	16.0
min # of input figures	0.0
ave. length of Experiment&Explanation list	5.7
ave. length per experiment (in words)	34.3
ave. length per explanation (in words)	27.1
max length of Experiment&Explanation list	13
max length per experiment (in words)	135
max length per explanation (in words)	89
min length of Experiment&Explanation list	2
min length per experiment (in words)	9
min length per explanation (in words)	9

Table 5: The statistics of ExpDesign .

- Mistral-7B: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3
- Mixtral-8x22B-MoE: https://huggingface.co/mistralai/Mixtral-8x22B-Instruct-v0.1
- Llama 3.1-70B: https://huggingface.co/meta-llama/Llama-3.1-70B
- Qwen 2.5-72B: https://huggingface.co/Qwen/Qwen2.5-72B

We use VLLM to unify the inference endpoints of all the above models.[10] We use Pytorch 2.4.0 with CUDA 12.1, and use 8 NVIDIA A100 GPUs for the LLMs inference.

# of instances	993
# of source papers	993
ave. input context length (in words)	9,811
max input context length (in words)	49,195
min input context length (in words)	24
ave. # of input figures	7.0
max # of input figures	37.0
min # of input figures	0.0
ave. # of input tables	4.3
max # of input tables	53.0
min # of input tables	0.0
ave. # of reviewers per paper	3.8
max # of reviewers per paper	9.0
min # of reviewers per paper	3.0
ave. # of weaknesses per reviewer	4.8
max # of weaknesses per reviewer	39.0
min # of weaknesses per reviewer	1.0
ave. length of weakness (in words)	39.1
max length of weakness (in words)	371.0
min length of weakness (in words)	2.0

Table 6: The statistics of Weakness .

Meanwhile, we use the gpt-4o-2024-08-06, gpt-4-1106-preview, o1-preview-2024-09-12, gemini-1.5-pro-002, and claude-3-5-sonnet-20240620 for the closed-source LLMs. We use LiteLLM to unify the API calling for all these LLMs.[11]

Given the unstable performance of LLMs, particularly closed-source ones, we run each model thrice during our experiments, selecting the median result from these repeated runs.

More Experiment Results

Input Context Scaling Investigation

Figure 4, Figure 5, and Table 7 show the context scaling results of EqInfer, ExpDesign, and Weakness.

Two Different Explanation Generation Methods for LLMs

We post the explanation generation performance comparison of containing self-containment in Table 8.

Human Evaluation on LLM-Generated Explanation

We ask 5 annotators to evaluate the LLM-generated explanations. Specifically, each of them is assigned 4 or 5 papers, along with the corresponding experiment lists. For each paper, the annotator is given 5 different models’ outputs (model names are anonymized), and the annotator has to decide if

[10] https://github.com/vllm-project/vllm

[11] https://github.com/BerriAI/litellm

Models	Input Context Processing	Window Size (in words)	SN-F1	SN-Precision	SN-Recall	ITF-IDF
GPT-4-Turbo	split-combine	3,000	**47.66**	42.15	**55.19**	5.31
	no-split	3,000	45.80	**43.66**	48.39	**5.58**
	no-split	20,000	44.99	42.64	47.82	**5.58**
GPT-4o	split-combine	3,000	**47.73**	42.09	**55.48**	5.95
	no-split	3,000	45.74	**43.45**	48.54	5.92
	no-split	20,000	45.47	42.97	48.51	**6.02**
AI-SCI	split-combine	3,000	**45.05**	40.02	**51.91**	2.23
	no-split	3,000	42.56	**40.90**	44.65	2.53
	no-split	20,000	42.53	40.75	44.78	**2.58**

Table 7: The performance comparison of different input processing methods for Weakness. We use GPT-4o and GPT-4-Turbo because both accept a maximum of 128k tokens input. We also put the results of AI-SCI in the table for reference. Here, “split-combine” splits the input paper into several pieces, where each piece’s length is denoted as “window size”; “no-split” means the conventional input cutting, for example, if the window size is 3,000, then only the first 3,000 words in the paper are used. According to the data statistics, 20,000 words can cover maximum lengths of more than 95% of the papers in our dataset.

Models	One-by-One	Whole-List
Llama 3.1-70B	50.05	49.36 (↓ 0.7)
Qwen 2.5-72B	51.12	48.56 (↓ 2.6)
Gemini 1.5 Pro	52.87	57.48 (↑ 4.6)
Claude 3.5 sonnet	53.03	59.11 (↑ 6.1)
GPT-4	55.03	56.95 (↑ 1.9)
GPT-4o	54.79	58.54 (↑ 3.8)
o1-preview	58.55	61.58 (↑ 3.0)

Table 8: The impact on S-Match scores of maintaining the experiment’s self-containment for ExpDesign.

Models	Acc. ratio
Llama 3.1-70B	22.93
Gemini 1.5 Pro	55.07
Claude 3.5 sonnet	61.46
GPT-4o	69.72
o1-preview	**76.14**

Table 9: The human evaluation results on LLMs’ output explanations of ExpDesign. “Acc. ratio” means how many model outputs are accepted by the annotator.

each LLM-generated explanation is acceptable according to the experiment. We show the human evaluation results in Table 9,

Multi-Modal Input Ablation

We post the multi-modal ablation study of ExpDesign and Weakness in Table 10 and Table 11.

Data cases and Annotation Platform Illustration

As shown in Figure 7, 8, and 9, we show the sample cases of the three tasks in AAAR-1.0. Meanwhile, we illustrate the screenshot of our annotation platform in Figure 6.

Prompt Templates

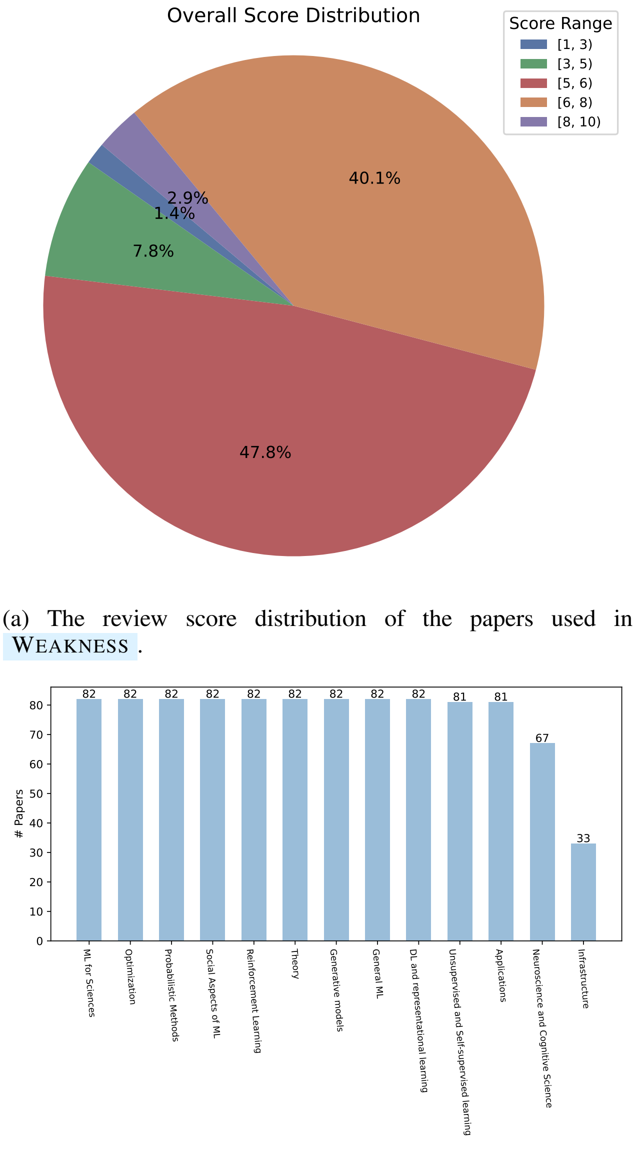
In this appendix, we attach all the prompts used in this work, including prompts in data collection and model prediction, as shown in Figure 10, 11, and 12.

Models	Experiment Design			Motivation Explanation		
	S-F1	S-Precision	S-Recall	S-Match	ROUGE-L	ROUGE-1
GPT-4o	**53.00**	**51.24**	**55.12**	**58.54**	**29.25**	**35.50**
w/ figures	50.11	48.94	51.59	58.53	27.87	34.30
GPT-4	**43.89**	42.34	**45.82**	**56.95**	**25.98**	**33.37**
w/ figures	43.54	**42.56**	44.85	55.03	22.82	30.01
InternVL2-26B	**40.52**	**48.95**	**35.20**	50.03	29.13	**34.26**
w/ figures	38.83	46.91	33.70	**50.29**	**29.29**	34.06

Table 10: The figure inputs ablation of ExpDesign. For the maximum text input length, same as the setting in Table 2, we use 2,000 and 3,000 words for open- and closed-source models, respectively. For the closed-source GPT-4o and GPT-4, as they have long context window, we use all the figures of each paper. While for InternVL2, we randomly select two figures per input paper.

Models	SN-F1	SN-Precision	SN-Recall	ITF-IDF
GPT-4o	**47.73**	**42.09**	**55.48**	**5.95**
w/ tables	46.76	41.32	54.17	5.53
w/ figures	46.62	41.20	54.04	5.48
w/ tables & figures	46.58	41.17	53.98	5.36
InternVL2-26B	41.91	41.02	43.28	**1.48**
w/ tables	40.55	40.37	42.91	1.46
w/ figures	**42.88**	**42.10**	**43.76**	1.46
w/ tables & figures	42.44	42.00	43.31	1.44

Table 11: The ablation study about the paper tables and figures of Weakness. Based on the conclusion in Table 7, we use the “split-combine” to process the text input here (2,000 and 3,000 words context window size for open- and closed-source models). For GPT-4o, we use all the table/figure images; while for InternVL2, we randomly select two images per paper, i.e., two random figures, two random tables, or one random figure + table.



(a) The review score distribution of the papers used in WEAKNESS.

(b) The track distribution of the papers used in WEAKNESS.

Figure 3: The data diversity illustration of WEAKNESS, including the score distribution and track distribution of the papers used in our dataset.

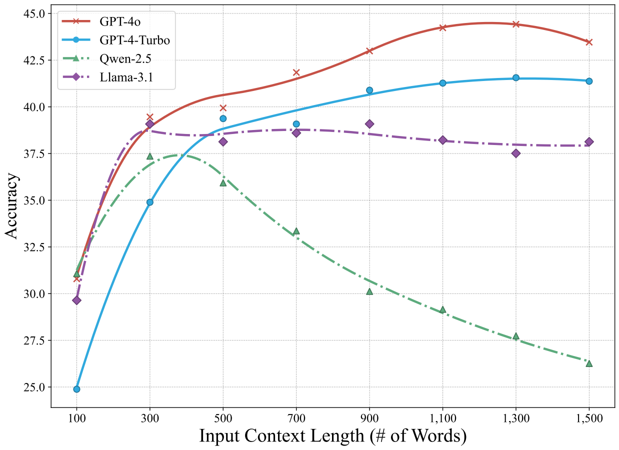


Figure 4: The input context length scaling trend on the EQINFER task.

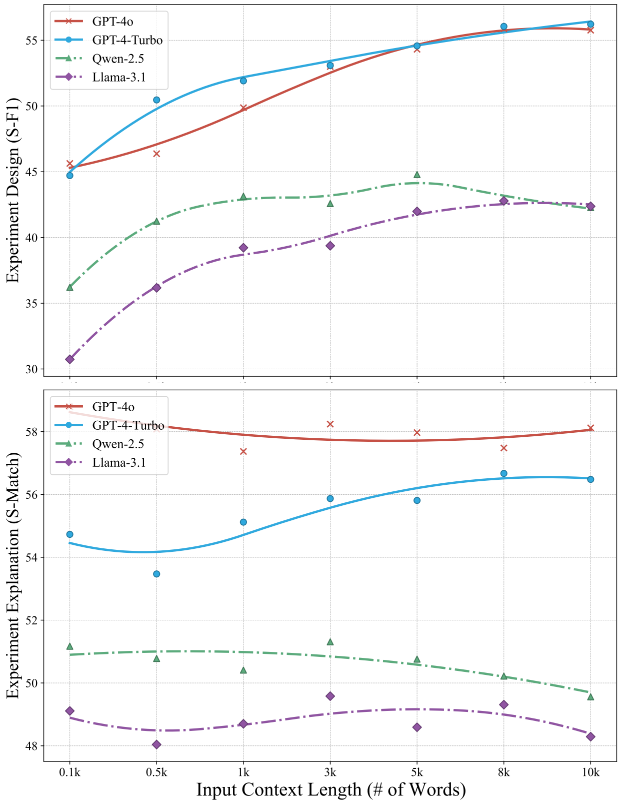
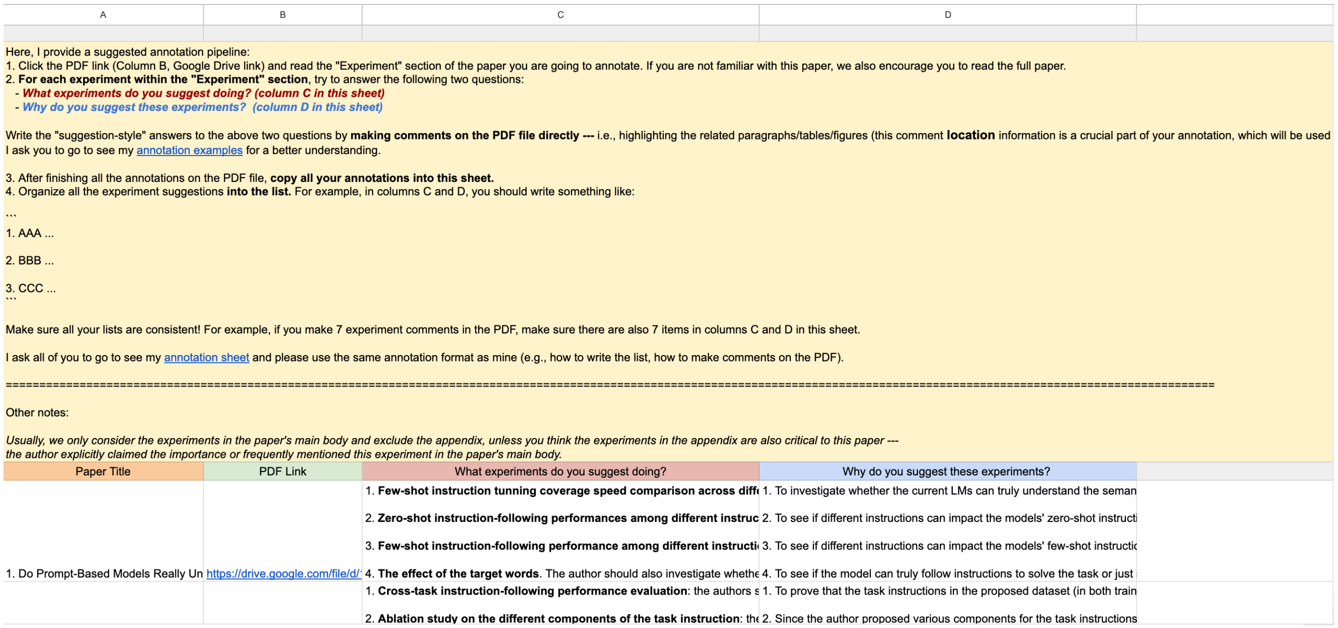


Figure 5: The input context length scaling trend of different LLMs on the EXPDESIGN task.



Here, I provide a suggested annotation pipeline:
1. Click the PDF link (Column B, Google Drive link) and read the "Experiment" section of the paper you are going to annotate. If you are not familiar with this paper, we also encourage you to read the full paper.
2. **For each experiment within the "Experiment" section**, try to answer the following two questions:
  - ***What experiments do you suggest doing? (column C in this sheet)***
  - ***Why do you suggest these experiments? (column D in this sheet)***

Write the "suggestion-style" answers to the above two questions by **making comments on the PDF file directly ---** i.e., highlighting the related paragraphs/tables/figures (this comment **location** information is a crucial part of your annotation, which will be used
I ask you to go to see my annotation examples for a better understanding.

3. After finishing all the annotations on the PDF file, **copy all your annotations into this sheet.**
4. Organize all the experiment suggestions **into the list.** For example, in columns C and D, you should write something like:

```
1. AAA ...

2. BBB ...

3. CCC ...
```

Make sure all your lists are consistent! For example, if you make 7 experiment comments in the PDF, make sure there are also 7 items in columns C and D in this sheet.

I ask all of you to go to see my annotation sheet and please use the same annotation format as mine (e.g., how to write the list, how to make comments on the PDF).

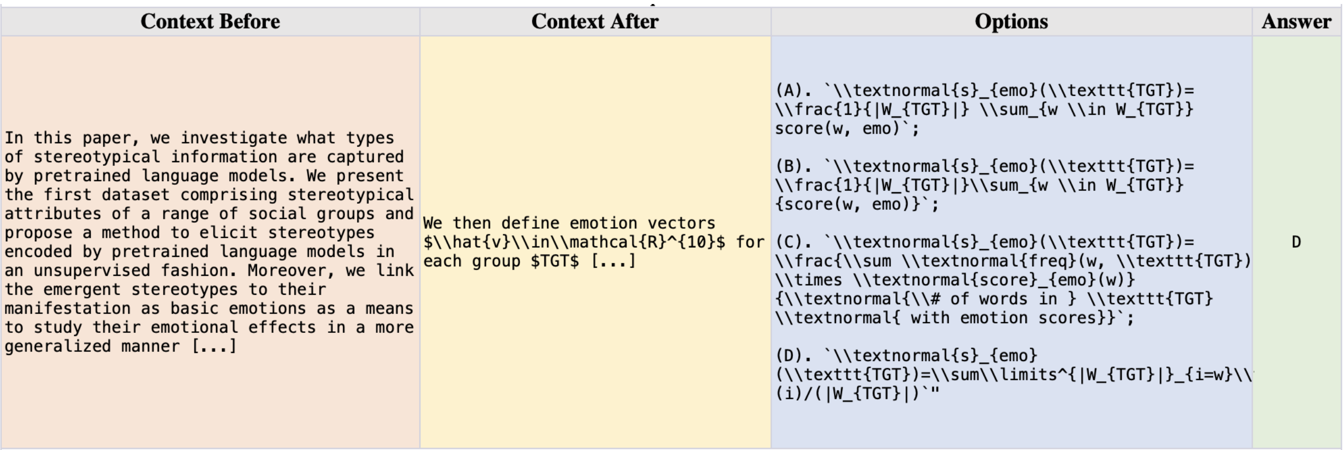
==

Other notes:

Usually, we only consider the experiments in the paper's main body and exclude the appendix, unless you think the experiments in the appendix are also critical to this paper ---
the author explicitly claimed the importance or frequently mentioned this experiment in the paper's main body.

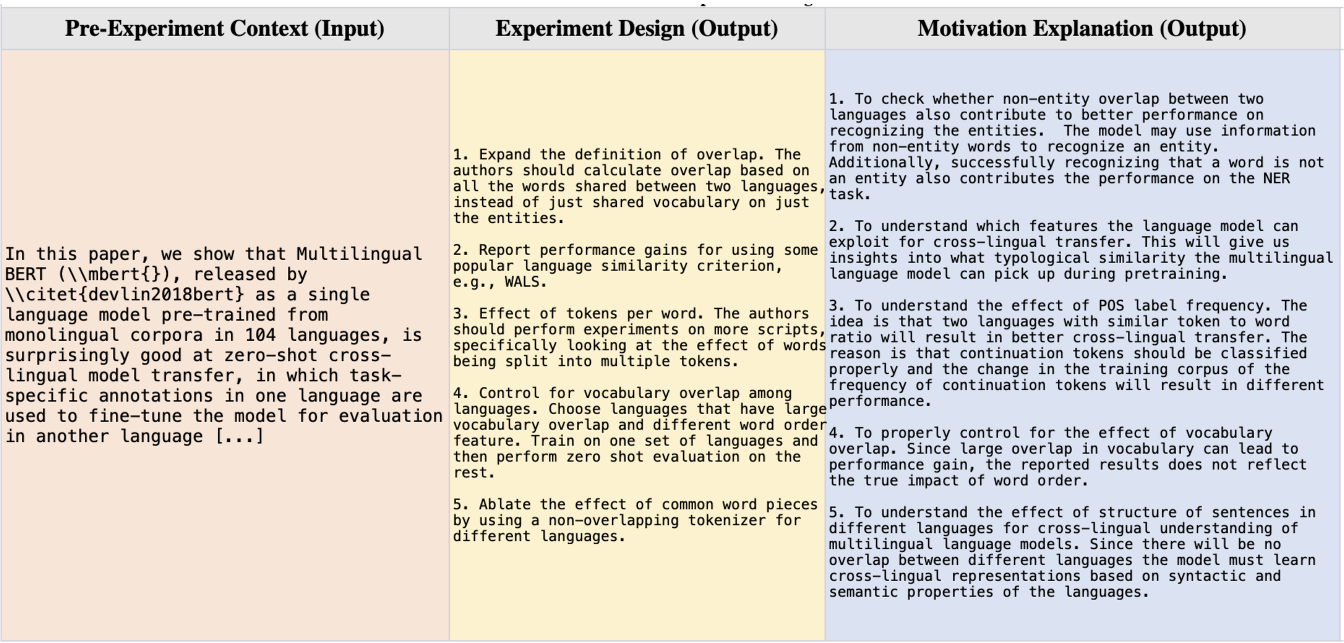
Paper Title	PDF Link	What experiments do you suggest doing?	Why do you suggest these experiments?
1. Do Prompt-Based Models Really Un	https://drive.google.com/file/d/1	1. **Few-shot instruction tunning coverage speed comparison across diff** 2. **Zero-shot instruction-following performances among different instruc** 3. **Few-shot instruction-following performance among different instructi** 4. **The effect of the target words**. The author should also investigate whethe	1. To investigate whether the current LMs can truly understand the seman 2. To see if different instructions can impact the models' zero-shot instructi 3. To see if different instructions can impact the models' few-shot instructio 4. To see if the model can truly follow instructions to solve the task or just
		1. **Cross-task instruction-following performance evaluation**: the authors s 2. **Ablation study on the different components of the task instruction**: the	1. To prove that the task instructions in the proposed dataset (in both train 2. Since the author proposed various components for the task instructions

Figure 6: The annotation platform for collecting the annotation of EXPDESIGN. We ask annotators to first make comments on the Google Drive PDF, then move all the annotations to the online Google Doc (for further verification and discussion).



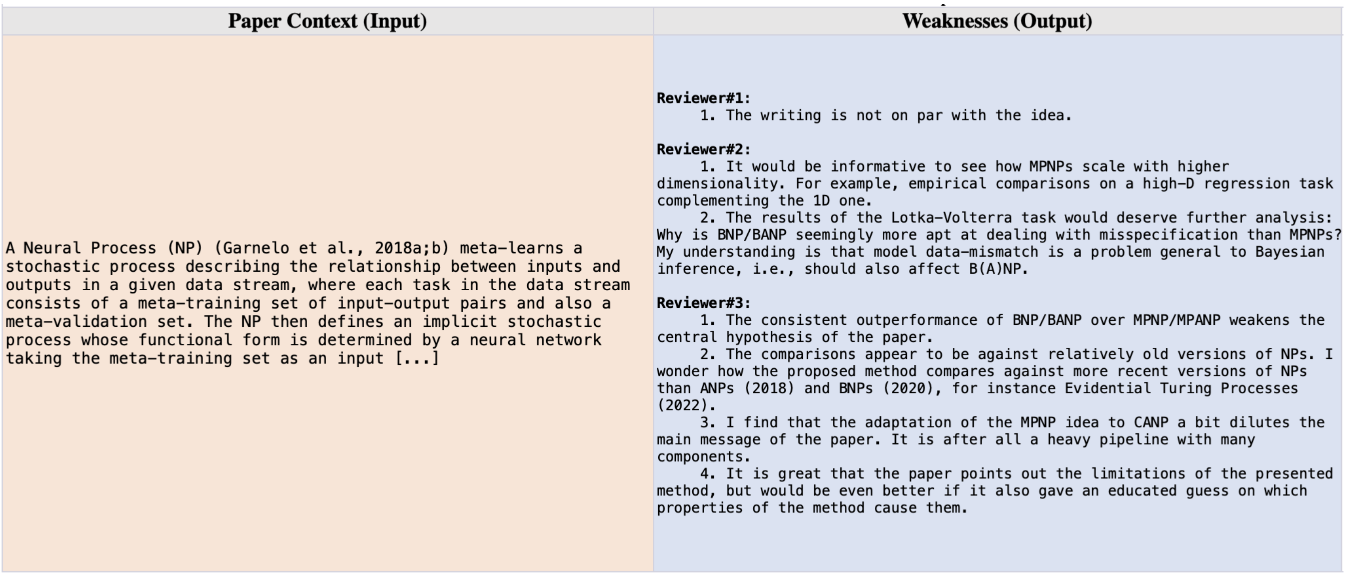
Context Before	Context After	Options	Answer
In this paper, we investigate what types of stereotypical information are captured by pretrained language models. We present the first dataset comprising stereotypical attributes of a range of social groups and propose a method to elicit stereotypes encoded by pretrained language models in an unsupervised fashion. Moreover, we link the emergent stereotypes to their manifestation as basic emotions as a means to study their emotional effects in a more generalized manner [...]	We then define emotion vectors $\\hat{v}\\in\\mathcal{R}^{10}$ for each group TGT [...]	(A). `\\textnormal{s}_{emo}(\\texttt{TGT})= \\frac{1}{|W_{TGT}|} \\sum_{w \\in W_{TGT}} score(w, emo)`; (B). `\\textnormal{s}_{emo}(\\texttt{TGT})= \\frac{1}{|W_{TGT}|}\\sum_{w \\in W_{TGT}} {score(w, emo)}`; (C). `\\textnormal{s}_{emo}(\\texttt{TGT})= \\frac{\\sum \\textnormal{freq}(w, \\texttt{TGT}) \\times \\textnormal{score}_{emo}(w)} {\\textnormal{\\# of words in } \\texttt{TGT} \\textnormal{ with emotion scores}}`; (D). `\\textnormal{s}_{emo} (\\texttt{TGT})=\\sum\\limits^{|W_{TGT}|}_{i=w}\\ (i)/(|W_{TGT}|)`"	D

Figure 7: A sample case of EQINFER.



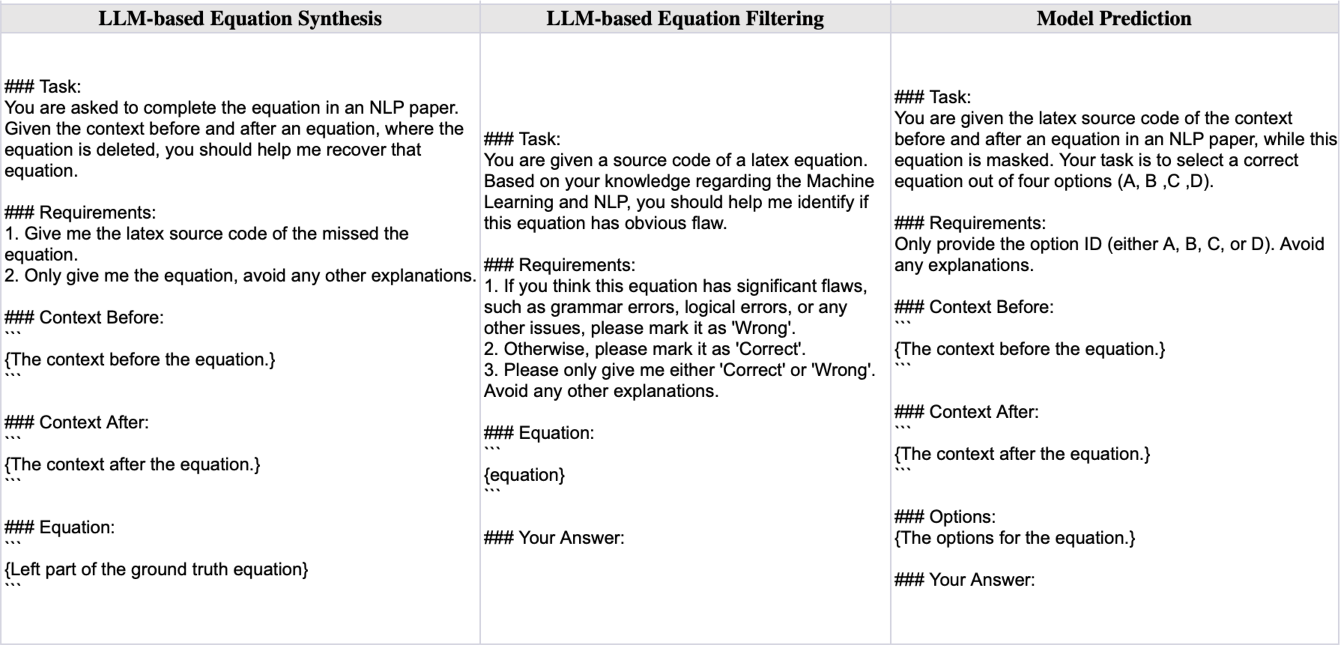
Pre-Experiment Context (Input)	Experiment Design (Output)	Motivation Explanation (Output)
In this paper, we show that Multilingual BERT (\\mbert{}), released by \\citet{devlin2018bert} as a single language model pre-trained from monolingual corpora in 104 languages, is surprisingly good at zero-shot cross-lingual model transfer, in which task-specific annotations in one language are used to fine-tune the model for evaluation in another language [...]	1. Expand the definition of overlap. The authors should calculate overlap based on all the words shared between two languages, instead of just shared vocabulary on just the entities. 2. Report performance gains for using some popular language similarity criterion, e.g., WALS. 3. Effect of tokens per word. The authors should perform experiments on more scripts, specifically looking at the effect of words being split into multiple tokens. 4. Control for vocabulary overlap among languages. Choose languages that have large vocabulary overlap and different word order feature. Train on one set of languages and then perform zero shot evaluation on the rest. 5. Ablate the effect of common word pieces by using a non-overlapping tokenizer for different languages.	1. To check whether non-entity overlap between two languages also contribute to better performance on recognizing the entities. The model may use information from non-entity words to recognize an entity. Additionally, successfully recognizing that a word is not an entity also contributes the performance on the NER task. 2. To understand which features the language model can exploit for cross-lingual transfer. This will give us insights into what typological similarity the multilingual language model can pick up during pretraining. 3. To understand the effect of POS label frequency. The idea is that two languages with similar token to word ratio will result in better cross-lingual transfer. The reason is that continuation tokens should be classified properly and the change in the training corpus of the frequency of continuation tokens will result in different performance. 4. To properly control for the effect of vocabulary overlap. Since large overlap in vocabulary can lead to performance gain, the reported results does not reflect the true impact of word order. 5. To understand the effect of structure of sentences in different languages for cross-lingual understanding of multilingual language models. Since there will be no overlap between different languages the model must learn cross-lingual representations based on syntactic and semantic properties of the languages.

Figure 8: A sample case of EXPDESIGN .



Paper Context (Input)	Weaknesses (Output)
A Neural Process (NP) (Garnelo et al., 2018a;b) meta-learns a stochastic process describing the relationship between inputs and outputs in a given data stream, where each task in the data stream consists of a meta-training set of input-output pairs and also a meta-validation set. The NP then defines an implicit stochastic process whose functional form is determined by a neural network taking the meta-training set as an input [...]	**Reviewer#1:** 1. The writing is not on par with the idea. **Reviewer#2:** 1. It would be informative to see how MPNPs scale with higher dimensionality. For example, empirical comparisons on a high-D regression task complementing the 1D one. 2. The results of the Lotka-Volterra task would deserve further analysis: Why is BNP/BANP seemingly more apt at dealing with misspecification than MPNPs? My understanding is that model data-mismatch is a problem general to Bayesian inference, i.e., should also affect B(A)NP. **Reviewer#3:** 1. The consistent outperformance of BNP/BANP over MPNP/MPANP weakens the central hypothesis of the paper. 2. The comparisons appear to be against relatively old versions of NPs. I wonder how the proposed method compares against more recent versions of NPs than ANPs (2018) and BNPs (2020), for instance Evidential Turing Processes (2022). 3. I find that the adaptation of the MPNP idea to CANP a bit dilutes the main message of the paper. It is after all a heavy pipeline with many components. 4. It is great that the paper points out the limitations of the presented method, but would be even better if it also gave an educated guess on which properties of the method cause them.

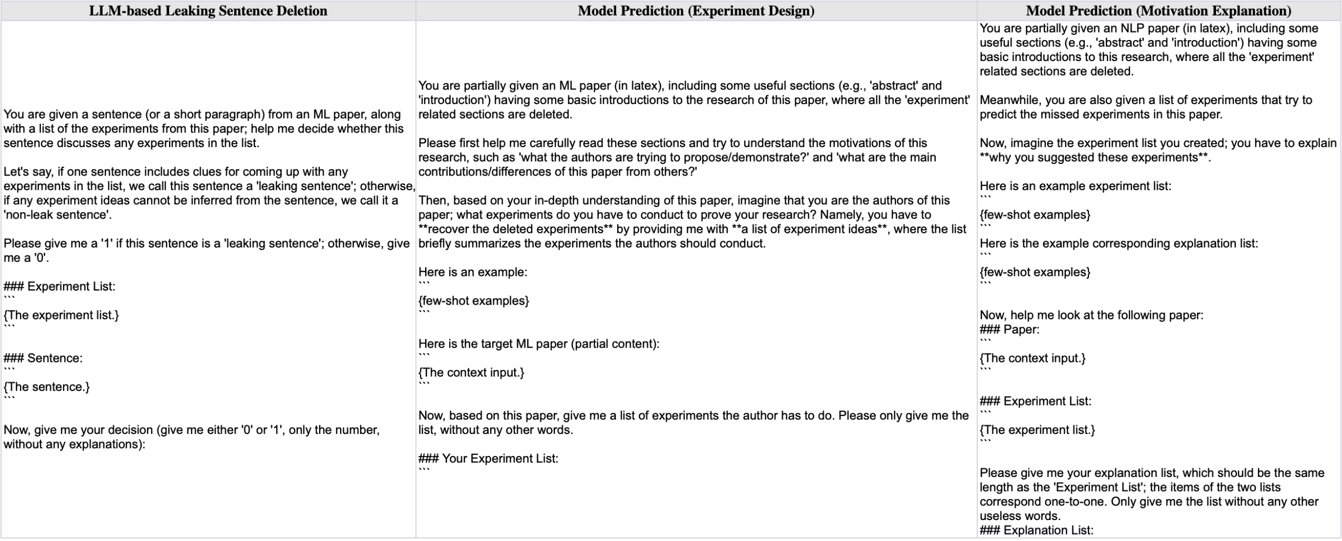
Figure 9: A sample case of WEAKNESS .



LLM-based Equation Synthesis	LLM-based Equation Filtering	Model Prediction

### Task: 	
You are asked to complete the equation in an NLP paper. Given the context before and after an equation, where the equation is deleted, you should help me recover that equation.
 	
### Requirements: 	
1. Give me the latex source code of the missed the equation.
2. Only give me the equation, avoid any other explanations.
 	
### Context Before: 	
```
{The context before the equation.}
```
 	
### Context After: 	
```
{The context after the equation.}
```
 	
### Equation: 	
```
{Left part of the ground truth equation}
```
	### Task:
	You are given a source code of a latex equation. Based on your knowledge regarding the Machine Learning and NLP, you should help me identify if this equation has obvious flaw.
	
	### Requirements:
	1. If you think this equation has significant flaws, such as grammar errors, logical errors, or any other issues, please mark it as 'Wrong'.
	2. Otherwise, please mark it as 'Correct'.
	3. Please only give me either 'Correct' or 'Wrong'. Avoid any other explanations.
	
	### Equation:
	```
	{equation}
	```
	
	### Your Answer:
		### Task:
		You are given the latex source code of the context before and after an equation in an NLP paper, while this equation is masked. Your task is to select a correct equation out of four options (A, B ,C ,D).
		
		### Requirements:
		Only provide the option ID (either A, B, C, or D). Avoid any explanations.
		
		### Context Before:
		```
		{The context before the equation.}
		```
		
		### Context After:
		```
		{The context after the equation.}
		```
		
		### Options:
		{The options for the equation.}
		
		### Your Answer:

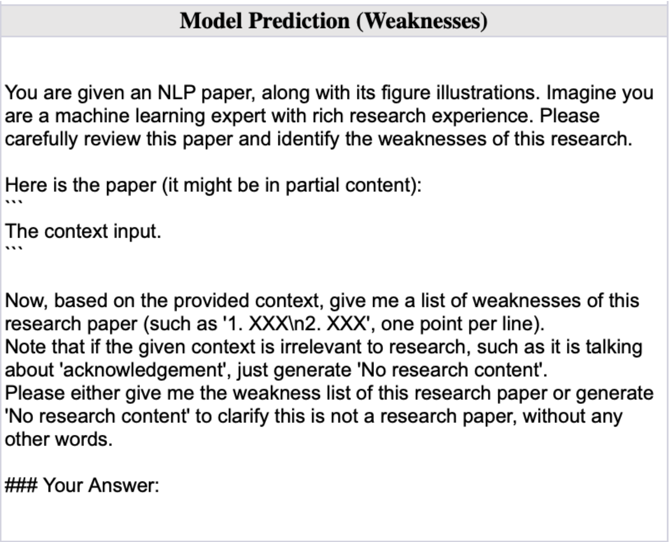
Figure 10: The prompts used in EQINFER, including both data collection and model prediction.



LLM-based Leaking Sentence Deletion	Model Prediction (Experiment Design)	Model Prediction (Motivation Explanation)

You are given a sentence (or a short paragraph) from an ML paper, along with a list of the experiments from this paper; help me decide whether this sentence discusses any experiments in the list.
 	
Let's say, if one sentence includes clues for coming up with any experiments in the list, we call this sentence a 'leaking sentence'; otherwise, if any experiment ideas cannot be inferred from the sentence, we call it a 'non-leak sentence'.
 	
Please give me a '1' if this sentence is a 'leaking sentence'; otherwise, give me a '0'.
 	
### Experiment List: 	
```
{The experiment list.}
```
 	
### Sentence: 	
```
{The sentence.}
```
 	
Now, give me your decision (give me either '0' or '1', only the number, without any explanations):
	You are partially given an ML paper (in latex), including some useful sections (e.g., 'abstract' and 'introduction') having some basic introductions to the research of this paper, where all the 'experiment' related sections are deleted.
	
	Please first help me carefully read these sections and try to understand the motivations of this research, such as 'what the authors are trying to propose/demonstrate?' and 'what are the main contributions/differences of this paper from others?'
	
	Then, based on your in-depth understanding of this paper, imagine that you are the authors of this paper; what experiments do you have to conduct to prove your research? Namely, you have to **recover the deleted experiments** by providing me with **a list of experiment ideas**, where the list briefly summarizes the experiments the authors should conduct.
	
	Here is an example:
	```
	{few-shot examples}
	```
	
	Here is the target ML paper (partial content):
	```
	{The context input.}
	```
	
	Now, based on this paper, give me a list of experiments the author has to do. Please only give me the list, without any other words.
	
	### Your Experiment List:
	``
		You are partially given an NLP paper (in latex), including some useful sections (e.g., 'abstract' and 'introduction') having some basic introductions to this research, where all the 'experiment' related sections are deleted.
		
		Meanwhile, you are also given a list of experiments that try to predict the missed experiments in this paper.
		
		Now, imagine the experiment list you created; you have to explain **why you suggested these experiments**.
		
		Here is an example experiment list:
		``
		{few-shot examples}
		```
		Here is the example corresponding explanation list:
		```
		{few-shot examples}
		```
		Now, help me look at the following paper:
		### Paper:
		```
		{The context input.}
		```
		### Experiment List:
		```
		{The experiment list.}
		```
		Please give me your explanation list, which should be the same length as the 'Experiment List'; the items of the two lists correspond one-to-one. Only give me the list without any other useless words.
		### Explanation List:
		```

Figure 11: The prompts used in EXPDESIGN, including both data collection and model prediction.



Model Prediction (Weaknesses)

~~~
You are given an NLP paper, along with its figure illustrations. Imagine you
are a machine learning expert with rich research experience. Please
carefully review this paper and identify the weaknesses of this research.

Here is the paper (it might be in partial content):
```
The context input.
```

Now, based on the provided context, give me a list of weaknesses of this
research paper (such as '1. XXX\n2. XXX', one point per line).
Note that if the given context is irrelevant to research, such as it is talking
about 'acknowledgement', just generate 'No research content'.
Please either give me the weakness list of this research paper or generate
'No research content' to clarify this is not a research paper, without any
other words.

### Your Answer:
~~~

Figure 12: The prompts used in Weakness .