Analysis of results in the ML research field

Investigating the Efficacy of LLMs in Extracting Stated Research Limitations

Alexia-Elena Predoi[1]
Supervisor(s): David M.J Tax[1], **Chenxu Hao**[1], **Hayley Hung**[1], **Nergis Tömen**[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Alexia-Elena Predoi
Final project course: CSE3000 Research Project
Thesis committee: David M.J. Tax, Chenxu Hao, Hayley Hung, Nergis Tömen, Klaus Hildebrandt

An electronic version of this thesis is available at https://gitlab.tudelft.nl/cse3000/analysisMLresults.

Abstract

The rapid growth of Machine Learning research has overwhelmed traditional peer-review systems, leading to concerns regarding reviewer fatigue and the consistency of scientific evaluation. While Large Language Models (LLMs) are being explored as potential assistants for quality assessment, their ability to objectively verify specific scientific criteria—such as those in the NeurIPS Paper Checklist—remains unproven. This checklist serves as a structured self-auditing framework that mandates authors to explicitly declare critical details, including potential negative societal impacts, exact hyperparameter tuning ranges, and clear definitions of theoretical assumptions or limitations. This study investigates the core question: "How well can an LLM extract the limitations described in scientific papers?" Using a manually annotated dataset of 78 papers, this research evaluates the accuracy of LLMs in extracting limitations stated by authors. Our findings reveal that while the LLM achieves perfect accuracy (100%) in detecting the presence of dedicated limitation sections, its performance in textual extraction is more nuanced. For explicit limitations, the model demonstrates high recall (0.91) but moderate precision (0.71), frequently over-extracting context. Furthermore, when tasked with extracting implicit limitations from papers lacking dedicated sections, both recall (0.71) and precision (0.69) decline. Notably, we found that a major bottleneck in unstructured text is getting the LLM to look at the actual weakness instead of getting distracted by subsequent sentences talking about future work. By comparing LLM performance against a human-verified ground truth, this work provides a feasibility study for automating high-stakes research quality assessments and identifies current bottlenecks in LLM reasoning for scientific auditing.

1 Introduction

The integrity of the peer-review process in Machine Learning (ML) is currently facing a scalability crisis. With conferences like NeurIPS receiving an unprecedented volume of submissions, the demand for qualified human reviewers has outpaced supply, resulting in severe reviewer fatigue and potentially inconsistent evaluations [3]. This strain has highlighted the urgent need for automated tools capable of assisting in the verification of research quality and adherence to standardized protocols.

To address these challenges, major venues have implemented structural interventions like the NeurIPS Paper Checklist to enforce transparency regarding reproducibility, claims, and limitations [6]. While Large Language Models (LLMs) are an obvious candidate to help automate these checks, recent real-world pilots show that AI assistants are highly vulnerable to "superficial agreement" and can easily be "gamed" by authors to generate false reassurance [1]. A significant research gap remains in understanding whether LLMs can objectively evaluate complex checklist criteria at a deep, sentence-level layer, or if they merely rely on shallow keyword matching.

The primary objective of this work is to determine how effectively an LLM can perform as a quality auditor. This is explored through the following research questions:

- **RQ1 (Explicit Auditing):** Can the LLM detect a dedicated limitations section and precisely extract the core limitation phrases while ignoring surrounding filler or future work?
- **RQ2 (Implicit Auditing):** In the absence of a dedicated section, can the LLM locate scattered or hidden limitation phrases throughout the full text?
- **RQ3 (Contextualization):** Does the LLM accurately explain why the identified factor constitutes a research limitation?

To address these questions, we design an automated evaluation framework that passes parsed manuscripts through a multimodal extraction engine, benchmarking its programmatic outputs directly against an expert-annotated human ground truth.

The remainder of this paper is structured as follows. Section 2 reviews relevant literature on peer-review automation. Section 3 formally defines the extraction task and evaluation framework. Section 4 details the architectural design and implementation of our LLM pipeline. Section 5 outlines the experimental setup and presents our empirical results for each research question. Section 6 addresses the ethical and reproducibility considerations of this work, followed by a broader interpretation of our findings in Section 7. Finally, Section 8 concludes the paper and discusses directions for future work.

2 Related Work

The automation of academic auditing and peer review has rapidly evolved from speculative natural language processing (NLP) tasks to deployed field experiments. Initial literature in this domain focused heavily on full-scale text generation to assist overburdened reviewers. For instance, Liu and Shah [5] introduced 'ReviewerGPT' to evaluate the feasibility of generating complete, multi-section peer reviews from raw manuscript inputs. While these generative approaches demonstrated a surface-level understanding of paper structures, subsequent studies raised critical concerns regarding the hallucination of technical flaws and an inability to verify underlying scientific truths [3].

As a result, contemporary research has shifted from broad review generation to targeted compliance auditing, specifically focusing on standardized meta-review frameworks like the NeurIPS Paper Checklist [6]. However, deploying LLMs for compliance tracking has revealed severe operational vulnerabilities and susceptibility to prompt manipulation. Documented field trials [1] revealed that deploying models as interactive author assistants resulted in systemic "superficial agreement," where authors could easily manipulate model prompts to bypass checklist constraints without actually correcting missing manuscript criteria. To mitigate this vulnerability to author gaming, subsequent work by Ye et al. [8] established that single-pass prompting is fundamentally unviable for high-stakes academic auditing, arguing instead for multi-stage pipelines governed by strict ethical and operational boundaries.

Beyond behavioral vulnerabilities, the literature highlights deep algorithmic reasoning gaps when LLMs handle complex scientific prose. Evaluating documents for structural consistency requires long-context, non-local textual reasoning. This bottleneck is explicitly highlighted by benchmarks such as CLAIM-BENCH, which demonstrated that state-of-the-art LLMs consistently fail to map abstract claims made in an introduction to the concrete empirical data presented in a results section [2].

While existing literature clearly proves that complex, structured pipelines are required to handle rigorous scientific verification [2, 8], a distinct gap remains. Current benchmarks focus on explicit macro-level relationship mapping (such as matching explicit claims to explicit figures). It remains unanswered how deeply an LLM can reason when tasked with extracting highly localized, context-dependent text elements—specifically, isolating explicit limitation statements from adjacent filler text, or scanning full manuscripts to uncover hidden, implicitly phrased limitations. While the LLM cannot verify the scientific validity of these limitations, localizing them within dense text significantly reduces reviewer fatigue by helping a human expert find them faster.

3 Problem Definition and Methodology

To evaluate how effectively Large Language Models (LLMs) can act as automated quality auditors, we developed a structured research methodology. This section defines the core information extraction task and explains our evaluation strategy for addressing each research question.

3.1 Extraction Task Definition

The primary task in this study is defined as *targeted limitation extraction*. Given a scientific manuscript, the goal of the automated auditor is to parse the document and extract literal, verbatim phrases where the authors acknowledge a weakness or constraint.

To evaluate this automated process, we establish a human-verified baseline across two distinct scenarios: papers with formal, explicitly labeled limitations sections, and papers where limitations are woven implicitly into general text. A team of human expert annotators manually reads the papers to identify and extract these phrases, providing a ground-truth benchmark to measure the LLM's auditing accuracy.

3.2 Prompt Design and Schema Enforcement

The prompts for both research questions were engineered to minimize LLM hallucinations and enforce strict adherence to the extraction task, utilizing targeted strategies to handle the differing document structures of the two research questions.

Core Prompting Architecture (RQ1). For the explicit extraction task, the prompt relies on a combination of persona alignment, strict string boundaries, and structured data serialization. First, the prompt assigns the model the persona of an "expert peer-reviewer for the NeurIPS conference" to align its internal weights with academic critique. Within this framework, we explicitly define what constitutes a limitation ("a statement where the authors acknowledge a weakness, constraint, or boundary") and outline strict negative constraints. Because LLMs naturally tend to summarize or paraphrase text, the prompt strictly commands the model to extract the "exact textual statement... verbatim from the paper" and explicitly forbids paraphrasing to enable rigorous empirical evaluation. Finally, rather than prompting the model to output a markdown list and attempting to parse the string downstream, we utilize the API's `response_schema` configuration. This forces the model to return a deterministic JSON object with predefined keys, including `has_limitations_section`, `limitations_section_name`, and `limitations`. This design choice eliminates parsing errors and guarantees that the output can be directly evaluated by programmatic scripts. The complete prompt template utilized for this explicit baseline is provided in Appendix A.1.

Advanced Constraints for Unstructured Text (RQ2). When auditing papers lacking dedicated sections, the baseline prompt is augmented with heavy lexical and semantic constraints to mitigate the heightened risk of false positives; the comprehensive prompt specification for this task can be reviewed in Appendix A.2.

Under this paradigm, the prompt introduces an absolute requirement for explicit linguistic framing; the model is instructed to only extract statements containing clear identifying phrases, such as "A weakness of our approach is..." or "Our method is limited by...". This boundary prevents the LLM from inappropriately labeling standard factual reports or scope definitions as self-acknowledged limitations.

Furthermore, the RQ2 prompt introduces contextual disambiguation instructions to handle complex narrative structures. Authors frequently describe a limitation in one sentence and refer back to it in the next (e.g., "This limitation suggests a direction for future research."). To ensure high-quality extraction, the model is explicitly instructed to resolve this co-reference by extracting the substantive sentence containing the actual weakness, rather than the referential sentence pointing to future work. Finally, to curb hallucinations in unstructured text, we introduced an expanded negative constraint module under a "Critical Instruction" banner that explicitly forbids the model from inferring or guessing limitations. The prompt actively forces the pipeline to filter out general discussions of sub-optimal hyperparameters, omitted metrics, or the background limitations of prior work, unless the authors explicitly frame them as an inherent weakness of their own study.

4 Approach and Design

To rigorously evaluate Large Language Models (LLMs) on the task of automated limitation extraction, we engineered a fully automated processing pipeline. This section details the architecture of the pipeline and provides justifications for the key design choices regarding model selection, prompt engineering, and evaluation parsing strategies.

4.1 Pipeline Components

The evaluation pipeline is implemented in Python to ensure open-source reproducibility within the machine learning community, and it consists of three sequential modules:

1. **Pre-processing and Purification:** The pipeline first ingests raw PDF submissions. Using the PyMuPDF library, it scans for the boilerplate "NeurIPS Paper Checklist" and dynamically truncates the document prior to this section. This prevents the LLM from inadvertently parsing checklist explanations as limitations.

2. **LLM Extraction Engine:** The purified PDF is uploaded directly to the LLM via its native File API. The model processes the document according to a carefully engineered prompt and outputs the extracted data directly into a strict JSON schema containing the section name, an existence flag, and a list of verbatim limitation quotes.

3. **Deterministic Evaluation Module:** A custom Python script parses the generated JSONs and evaluates them against the human-annotated ground truth. It normalizes the extracted text and categorizes the outputs into exact matches, omissions, hallucinations, and segmentation variances.

4.2 Model Selection and Multimodal Ingestion

We selected the *Gemini 3 Flash Preview* model as the core extraction engine. The primary justification for this choice over traditional text-based LLM architectures (e.g., standard GPT-3.5/4 text endpoints) is its native multimodal processing capabilities via the Google GenAI File API. Because scientific PDFs are inherently multimodal documents containing both visual layout elements, figures, and text, a traditional text-only extraction engine risks losing vital structural context. By utilizing a natively multimodal architecture, the pipeline feeds both the full PDF document and the textual execution prompt simultaneously to the model, allowing it to perform its analysis directly on the raw visual layout.

Scientific papers frequently utilize complex multi-column layouts, floating figures, and varied typographic hierarchies (e.g., bolded subsection headers). Raw text extraction flattens this structure, often destroying the boundaries of the "Limitations" section and merging it with unrelated text. By uploading the raw PDF file directly, the Gemini model utilizes its vision encoder to "see" the document exactly as a human reviewer would. This multimodal ingestion allows the model to leverage spatial and typographic cues to accurately locate specific subsections without relying on regular expression searches. Furthermore, the Flash variant was selected to balance high-context reasoning capabilities with rapid, cost-effective inference over large document batches.

4.3 Parsing and Evaluation Strategy

Comparing model-extracted quotes against human annotations is tricky because humans and AI rarely choose the exact same sentence boundaries (e.g., one might include a word like "However," at the start while the other cuts it out). We call these *chunking variances*.

To handle this fairly, we use a simple text-normalization and word-matching process:

1. **Text Normalization:** We convert all extracted text and human annotations to lowercase, strip out

all punctuation, and remove extra whitespace.

2. **Word Intersection:** Instead of checking for a 100% perfect sentence match, our script checks for overlapping words between the two text sets.

We chose this exact matching approach over standard evaluation metrics like ROUGE [4] or BLEU [7] because our task requires verifying factual, verbatim quotes rather than evaluating creative text generation. This strategy lets us accurately sort outputs into true matches, complete omissions, or hallucinations. Crucially, it allows us to automatically flag "differently chunked" lines so the model isn't unfairly penalized for minor boundary discrepancies.

5 Experiments

This section details the empirical evaluation environment and presents the quantitative performance results across two of our core research questions.

5.1 Experimental Setup

Evaluation Corpus: Our dataset consists of a curated evaluation corpus of $n = 78$ peer-reviewed research papers accepted to the NeurIPS main conference track.

Execution Environment: The pipeline was developed in Python 3.9 using the `PyMuPDF` library for structural document trimming, querying the `gemini-3-flash-preview` engine. To ensure deterministic outputs and guarantee experimental reproducibility, the model's temperature parameter was explicitly set to 0.0.

5.2 RQ1: Explicit Auditing Results

To answer our first research question regarding the model's ability to explicitly audit NeurIPS submissions, we evaluated the pipeline's performance across two dimensions: macro-level section detection and micro-level textual extraction precision, along with an analysis of formatting variances.

Binary Section Detection and Textual Extraction

At the macro-level, the model's ability to determine whether a paper explicitly contained a limitations section was highly robust. Evaluated against the human ground truth across the 78-paper dataset, the multimodal pipeline achieved a perfect section detection accuracy of 100.0%. It successfully identified all 48 papers containing a limitations section (100.0% Recall) and correctly dismissed the 30 papers without one, producing zero hallucinations (100.0% Precision). Furthermore, out of the 48 positive detections, the model extracted the exact section header name correctly in 46 instances (95.8% accuracy). These metrics demonstrate that the model can successfully navigate complex, multi-column scientific layouts and reliably identify structural compliance directly from the document layout.

In Figure 1, the LLM extraction performance is shown. The recall is 0.89, correctly extracting 106 true positive limitations and missing only 13 explicit limitations (False Negatives) out of the 137 total ground truth limitations. When the model did omit a limitation, qualitative review showed it was typically because the authors buried the constraint deep within an unrelated paragraph or used implicit phrasing rather than formal limitation framing.

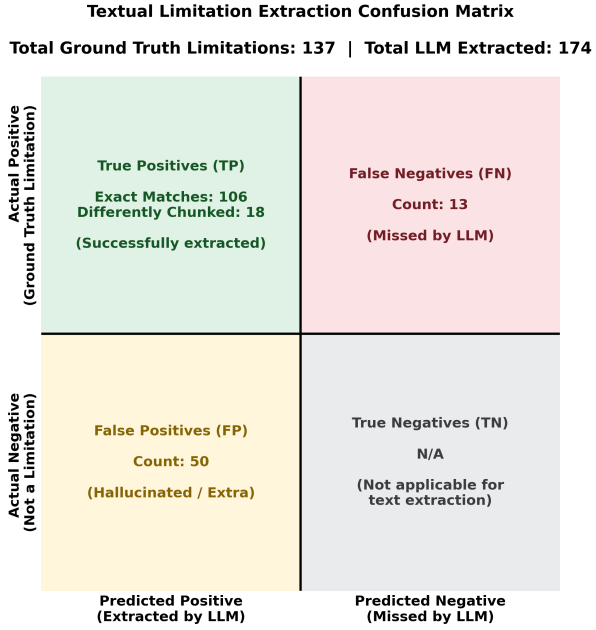

Figure 1: Confusion matrix detailing the micro-level performance metrics for textual limitation extraction, including true positives, omissions (false negatives), and hallucinations (false positives).

Conversely, Extraction Precision suffered (0.68) due to a significant tendency to over-extract. The evaluation script identified 50 False Positives (hallucinated extractions). A manual review of these False Positives revealed a clear behavioral pattern: the LLM consistently conflates proposed future work or minor system artifacts with formal limitations. For example, in one paper, the model extracted *"Human-level dexterity, even in simulation, remains challenging"*—a generalized statement of future difficulty rather than a strict constraint of the proposed method. Despite explicit prompt instructions to ignore future work, the model's high sensitivity to "limitation-adjacent" rhetoric causes it to aggres-

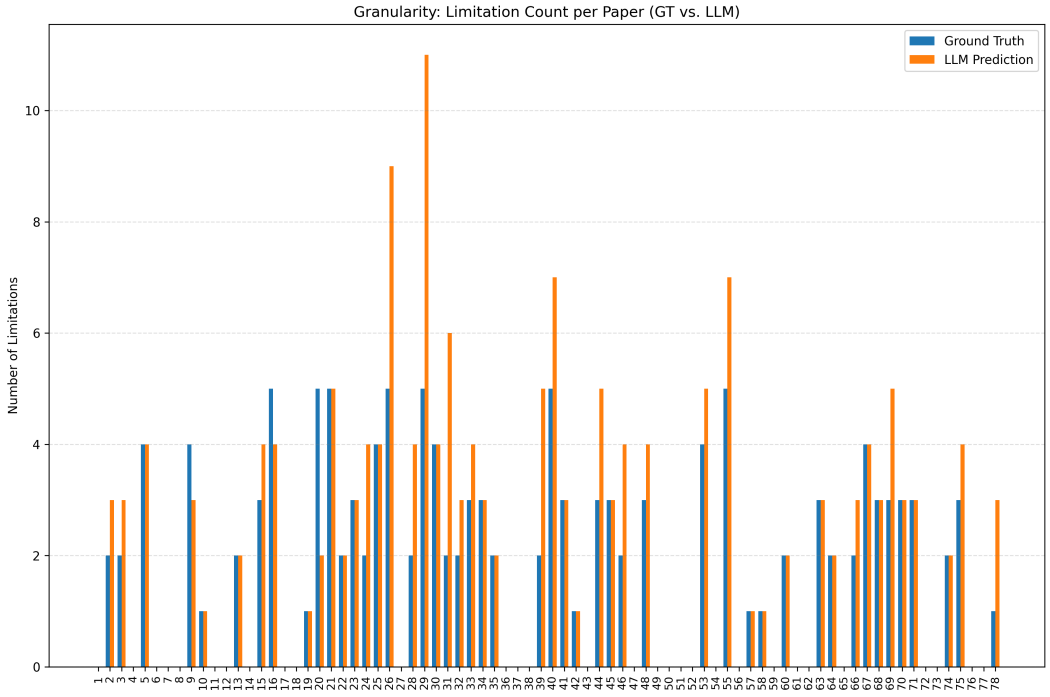

Figure 2: Granularity alignment matrix mapping the total number of human-annotated ground truth limitations against the model-extracted limitation counts across all 78 evaluated papers.

sively pull these statements. This indicates that while LLMs are excellent at retrieving all potential weaknesses (high recall), their zero-shot precision must be carefully constrained to perfectly align with strict human-auditing standards.

The Impact of Segmentation Variances

Figure 2 plots the document-by-document volume discrepancy between the human ground truth and the LLM predictions. As shown in the chart, 54 out of the 78 papers (69.2%) shared the exact same limitation count between the ground truth and the LLM prediction. However, overall, the LLM extracted 174 total statements compared to the 137 identified by human annotators. This variance demonstrates that beyond strict hits and misses, generative text extraction introduces distinct complexities in granularity and chunking.

A significant portion of this discrepancy stems from "Differently Chunked" statements. The evaluation isolated 18 distinct instances where the LLM successfully located the limitation text but segmented it differently than the human baseline. Often, this manifested as the LLM merging two distinct human-annotated limitations into a single large paragraph quote, or conversely, splitting a single human-annotated paragraph into multiple bulleted points. For instance, the LLM included list formatting (e.g., "(1)", "(2)") or leading conjunctions that human annotators explicitly truncated.

These segmentation variances underscore a fundamental challenge in using LLMs as automated auditors: while the underlying semantic information is successfully retrieved, the rigid syntactic boundaries expected in programmatic evaluation are often violated. This highlights why evaluating LLM extractions requires a flexible matching system rather than requiring a strict letter-for-letter match. A purely binary evaluation would unfairly penalize the model for minor formatting discrepancies even when it successfully retrieves the correct information.

5.3 RQ2: Implicit Auditing Results

To answer our second research question regarding the model's ability to extract implicit limitations—those stated in papers lacking a dedicated "Limitations" section—we evaluated the pipeline's performance across the remaining 30 papers in our evaluation corpus. In this scenario, the LLM is forced to read the entire document to identify limitations woven into general text, such as within the Conclusion, Discussion, or Appendix sections.

Full-Document Textual Extraction Performance

When forced to scan the entire document for implicit phrasing, the model's extraction capabilities showed a

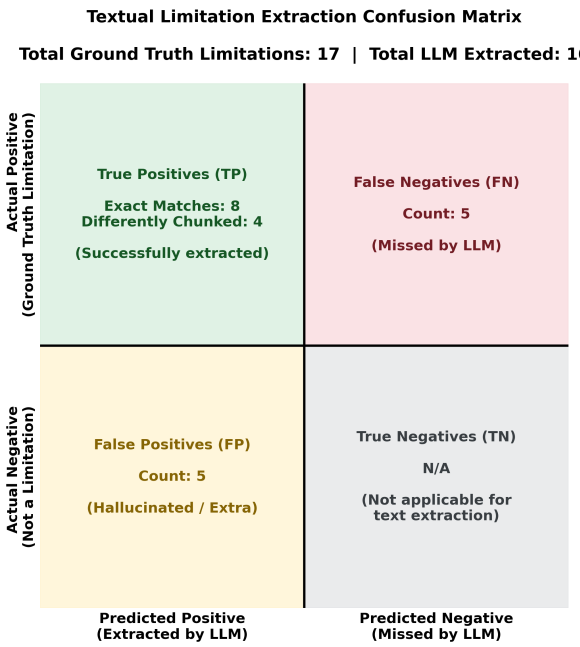

Figure 3: Confusion matrix mapping the textual limitation extraction performance on the 30-paper implicit auditing corpus (Total Ground Truth: 17 vs. Total LLM Extracted: 16).

noticeable decline compared to the explicit auditing task. The pipeline achieved an Extraction Recall of 0.71 and an Extraction Precision of 0.69.

The structural distribution of these extraction errors is illustrated in the confusion matrix in Figure 3. Out of a total of 17 ground truth limitations across this corpus, the model successfully isolated the semantic core of 12 instances, split between 8 exact matches and 4 differently chunked segments. However, the model suffered from a notable vulnerability to critical omissions, racking up 5 False Negatives (Missed claims) alongside 5 False Positives (Hallucinated/Extra extractions).

The drop in recall indicates that the model struggles to surface constraints when they are buried within dense, unstructured paragraphs. For example, in one paper, human annotators identified a hardware constraint located deep in the Appendix ("Note that we were not able to collect results for DAVINZ... due to hardware limitations"). The LLM entirely missed this statement, likely because it was visually and structurally isolated from the main concluding remarks and lacked explicit limitation framing.

Similarly, the Extraction Precision of 0.69 reflects a challenge with over-extraction and false positives. Without a rigid section boundary to act as a contextual guardrail, the LLM is significantly more susceptible to misinterpreting general experimental observations as formal limitations. For instance, in a paper which contained no formal limitations according to the ground truth, the model hallucinated an extraction by pulling a standard result-reporting sentence: "However, Causal CPC does not outperform other models in short-term forecasting, a consistent limitation across experiments." This demonstrates that while the model remains highly sensitive to negative performance phrasing, its zero-shot reasoning is prone to conflating standard evaluation metrics with explicitly acknowledged systemic weaknesses.

The Challenge of Implicit Framing and Segmentation

This behavioral instability becomes clear when analyzing the extraction volume on a paper-by-paper basis, as plotted in Figure 4. The model perfectly aligned with the ground truth count across the vast majority of the 30 documents, especially in identifying papers with zero limitations. However, systemic tracking errors emerge in documents with denser text profiles. In Paper 8 and Paper 28, the LLM omitted valid items, while in Paper 3 and Paper 30, it over-extracted, culminating in a granularity mismatch in Paper 30, where it predicted 3 limitations against a ground truth of 2.

The absence of a standardized limitations section also amplifies the segmentation variances observed in RQ1. When authors lack a dedicated section, they frequently aggregate multiple implicit limitations into complex, flowing narratives. When the LLM attempts to extract these, it often struggles to determine the correct syntactic boundaries of the claim.

A prime example of this chunking discrepancy occurred in one paper where human annotators separated the authors' constraints into two distinct, numbered limitation statements (focusing on simplified structures vs. task settings). The LLM, however, extracted the entire introductory clause alongside both points as a single, massive block of text: "Although we have made efforts in understanding ICL, there are still some limitations in our analysis: (1) our work primarily focuses on... (2) for more tasks...".

These occurrences reaffirm that LLM-based auditing systems must account for granular mismatch, especially in unstructured text. While the semantic core of the limitation is successfully retrieved from the depths of the paper, the model's propensity to either over-segment or merge multi-part implicit constraints necessitates the use of flexible text-matching criteria rather than strict binary string comparisons.

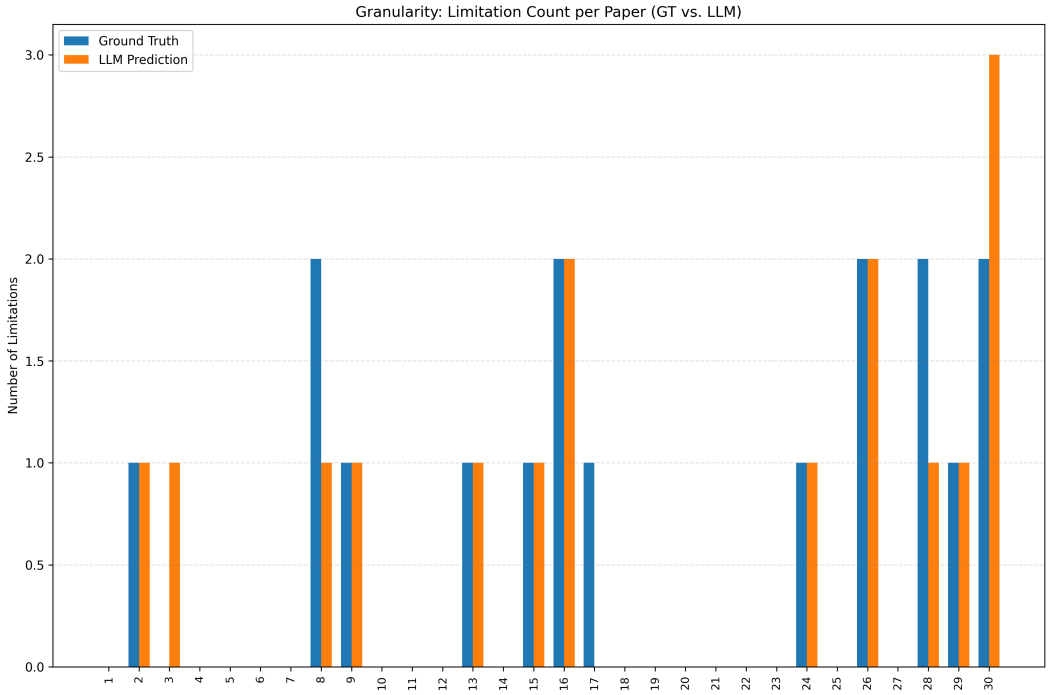

Figure 4: Granularity tracking chart plotting the human-annotated ground truth limitation counts against the automated LLM predictions on a per-paper basis across the 30-document implicit corpus.

6 Responsible Research

This section outlines the ethical considerations and reproducibility of the methods used in this project, ensuring alignment with responsible research practices.

6.1 Ethical Considerations

The primary ethical consideration of this research is the fair and accurate representation of the authors' original work. Our methodology is designed with several safeguards to ensure we analyze and present the data responsibly:

- **Data Source:** The analysis is performed on publicly available research papers from the NeurIPS 2024 conference. No private, confidential, or sensitive data is utilized in this study.
- **Purpose of Analysis:** The goal of this project is a meta-scientific study of how limitations are reported in machine learning literature. It is not designed to single out or critique individual papers or authors, but rather to aggregate results to highlight community-wide reporting patterns.
- **Minimizing Misrepresentation:** We have taken significant steps in our prompt engineering to minimize the risk of the Large Language Model (LLM) misinterpreting or hallucinating limitations. The LLM is explicitly instructed to extract the *exact verbatim text* of a limitation, preventing paraphrasing or summarization that could alter the authors' original meaning. Furthermore, the prompt provides a clear definition of what constitutes a "limitation" and establishes strict negative constraints to prevent over-extraction.
- **Use of AI:** We acknowledge the broader ethical questions surrounding the use of large-scale LLMs, including their environmental footprint and potential biases. Our application is narrow and focused on a specific, verifiable academic task. By using strict JSON schema enforcement (`response_schema`), we constrain the model's generation to programmatic evaluation, reducing the potential for generating unstructured or harmful content.

6.2 Reproducibility

We have structured the project to ensure the findings are highly reproducible. The entire analysis pipeline, from data processing to evaluation, is codified in a series of programmatic scripts.

- **Codebase and Pipeline:** The custom repository contains all scripts required to run the analysis end-to-end. This includes pre-processing

(`checklist_trimmer.py`), explicit and implicit LLM extraction (`gemini_analyzer.py`, `gemini_implicit_analyzer.py`), and quantitative/qualitative evaluation (`evaluate_results.py`, `evaluate_implicit_results.py`).

- **Model and Prompts:** The specific model version used is hardcoded as `gemini-3-flash-preview` to prevent silent downstream updates from altering the behavior. The full prompts sent to the model are included directly in the scripts, ensuring that the exact same constraints can be executed by other researchers.

- **Data Availability:** To fully reproduce the results, the original PDF files of the research papers and the human-annotated ground truth files (e.g., `ground_truth_limitations.json`) are made available.

- **Mitigating Non-Determinism:** While LLM outputs have inherent variability, we mitigate this by employing "Strict JSON Schema Enforcement" via Pydantic models passed directly to the Gemini API. This forces the model to return deterministic, correctly-typed JSON objects, which eliminates downstream parsing errors and significantly increases the consistency of the results across multiple runs.

6.3 Declaration of AI usage

Generative AI was used strictly in a supporting capacity to assist with code optimization, debugging, and paper text rephrasing. All core research concepts, experimental designs, data annotations, and interpretations remain entirely the work of the authors.

7 Discussion

Our results provide a quantitative and qualitative look into the capabilities of Large Language Models for automated meta-scientific analysis. Overall, our findings align with broader observations in natural language processing: LLMs excel at targeted information extraction when constrained by explicit structural markers (as demonstrated by our high success rate in RQ1), but they struggle significantly with the pragmatic and contextual nuances of academic discourse when those markers are absent (RQ2).

Limitations of the Ground Truth

We identify two distinct challenges regarding the human-annotated ground truth used in this study:

First, the annotators lacked highly specialized domain expertise. The manual annotation was performed by the authors of this study. While we hold general backgrounds in computer science, we are not experts in every niche machine learning sub-field represented at NeurIPS. This varying familiarity inevitably introduced cognitive biases into our annotations. Consequently, what our evaluation logged as an LLM "hallucination" or "miss" might occasionally represent a valid alternative interpretation of highly advanced methodology.

Second, the classification task itself is inherently subjective. Distinguishing between a standard methodological trade-off, a factual report of experimental scope, and a true self-acknowledged weakness is a highly ambiguous linguistic task. Even with strict guidelines, deciding when a text passage crosses the line into a formal constraint requires a judgment call, meaning absolute consensus on boundaries is difficult to define.

Explanation of Results and the Chunking Problem

When comparing the LLM's extractions to our ground truth, we observed a significant discrepancy in granularity, which we term the "chunking problem." Humans tend to synthesize a cohesive paragraph into a single conceptual limitation, whereas the LLM frequently atomizes the same text into multiple. This cognitive difference explains why our exact-match textual evaluation metrics occasionally under-reported the model's true semantic accuracy. It highlights a broader context in NLP evaluation: syntactic matching is insufficient for assessing complex extraction tasks, and future pipelines must incorporate semantic similarity metrics to fairly grade the LLM.

Technical Limitations and API Quotas

Finally, the scale and speed of our methodology were constrained by practical, technical limitations. Relying on a commercial LLM API (Gemini) meant we were subject to strict daily usage quotas and rate-limiting. These bottlenecks restricted the total volume of papers we could process within our timeframe and severely limited our ability to conduct extensive prompt ablation studies or hyperparameter tuning. While the pipeline is programmatically robust, its reliance on a rate-limited external service presents a barrier to scaling this meta-analysis to the tens of thousands of papers published annually across all major AI conferences.

8 Conclusions and Future Work

Our evaluation reveals a clear divide between structured and unstructured text extraction in academic auditing. For RQ1, when authors explicitly group limitations into a dedicated section, strict JSON schema enforcement and expert-reviewer personas minimize hallucinations, yielding highly deterministic outputs. However, answering RQ2 highlights significant challenges in pragmatic comprehension; when limitations are implicitly woven into methodology or conclusion sections, the LLM struggles to differentiate between factual experimental scope, future work, and true self-acknowledged weaknesses. A major open issue identified during our qualitative analysis is the "chunking problem", where human annotators and LLMs segment text at different granularities. While a human might extract a single comprehensive paragraph, the LLM frequently splits it into distinct bullet points, complicating traditional precision and recall metrics.

To address these limitations, future work should transition from strict string matching to semantic evaluation metrics that grant partial credit for semantically identical extractions. Furthermore, conducting an in-depth linguistic analysis of authors' hedging strategies and syntactic cues could help refine prompts for better contextual understanding. To improve implicit extraction performance, future research should implement advanced prompting paradigms, such as a Chain-of-Thought (CoT) approach, forcing the model to explicitly reason about why a sentence qualifies as a limitation before extracting it. Finally, expanding this programmatic pipeline beyond the NeurIPS dataset to other scientific fields will test its cross-domain generalization and investigate whether domain-specific reporting styles impact extraction accuracy.

References

[1] Alexander Goldberg, Ihsan Ullah, Thanh Gia Hieu Khuong, Benedictus Kent Rachmat, Zhen Xu, Isabelle Guyon, and Nihar B. Shah. Usefulness of LLMs as an author checklist assistant for scientific papers: Neurips'24 experiment. `https://www.cs.cmu.edu/~csd-phd-blog/2025/llm-checklist-assistant/`, 2025. Carnegie Mellon University CSD PhD Blog.

[2] Shashidhar Reddy Javaji, Yupeng Cao, Haohang Li, Yangyang Yu, Nikhil Muralidhar, and Zining Zhu. Can AI validate science? benchmarking LLMs on claim → Evidence reasoning in AI papers. In Kentaro Inui, Sakriani Sakti, Haofen Wang, Derek F. Wong, Pushpak Bhattacharyya, Biplab Banerjee, Asif Ekbal, Tanmoy Chakraborty, and Dhirendra Pratap Singh, editors, *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2355–2379, Mumbai, India, December 2025. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

[3] J. Liang. Stateful peer review for sustainable ML ecosystems. *ResearchGate Preprint*, 2024.

[4] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[5] Ryan Liu and Nihar B. Shah. ReviewerGPT? an exploratory study on using large language models for paper reviewing. *arXiv preprint arXiv:2306.00622*, 2023.

[6] Neural Information Processing Systems. NeurIPS 2021 paper checklist guidelines. `https://neurips.cc/public/guides/PaperChecklist`, 2021. Accessed: 2026.

[7] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[8] Haizhou Ye, Hongyuan Du, Zhenwei Zhou, Mengdi Zhang, and Minseok Shin. How should we responsibly adopt LLMs in peer review? A baseline evaluation framework. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, August 2026.

A LLM Prompt Specifications

To ensure complete reproducibility of our pipeline, this section provides the exact textual templates used for both the explicit (RQ1) and implicit (RQ2) extraction tasks. Both prompts were executed using the `response_schema` configuration via the Google GenAI File API.

A.1 RQ1: Explicit Section Extraction Prompt

The following prompt was supplied to the *Gemini 3 Flash Preview* model for documents containing structured, dedicated limitation indicators.

> "You are an expert peer-reviewer for the NeurIPS conference. You are provided with a PDF of a research paper from the 2024 Main Conference Track. Please extract the exact quotes of the limitations explicitly stated by the authors specifically within a clearly stated "Limitations" section, an equivalently named section (like "Discussion and Limitations"), or a distinct "Limitations" subsection (e.g., a subsection within the "Discussion" section).
>
> If the paper does not have a clearly stated "Limitations" section or subsection, you must state so.
>
> Please include: explicitly stated limitations of the proposed methods, theoretical assumptions, failure modes, algorithmic constraints, computational constraints, or any other weaknesses acknowledged by the authors.
>
> Please do not include: future work suggestions (unless they explicitly state a current limitation), background limitations (limitations of prior work), or weaknesses not explicitly stated by the authors.
>
> A limitation is defined as a statement where the authors acknowledge a weakness, constraint, or boundary of their proposed work or experimental setup.
>
> For each limitation, you must identify its ID (a sequential integer starting from 0) and the exact textual statement it makes, verbatim from the paper. Do not paraphrase, summarize, or alter the original text.
>
> You must also extract the exact name of the main section where you found these limitations. If they are located in a subsection, provide the name of the main parent section (e.g., if found in a "Limitations" subsection within the "Discussion" section, you must extract "Discussion").
>
> You must extract the exact, full title of the paper directly from the first page of the provided PDF and use it for the "paper_title" field.
>
> Return the output in the following format:

```
{
    "paper_title": "Exact title extracted from
        the PDF",
    "has_limitations_section": true,
    "limitations_section_name": "Name of the
        main section",
    "limitations": [
        {
            "limitation_id": 0,
            "limitation_text": ""
        }
    ]
}
```

> If there is no clearly stated Limitations section or subsection, set "has_limitations_section" to false, set "limitations_section_name" to null, and leave the "limitations" list empty.
>
> You MUST return only the valid JSON object with no additional text, explanations, or formatting."

A.2 RQ2: Implicit Auditing and Lexical Constraint Prompt

The following prompt was deployed across the unstructured document corpus where no formal structural section indicators were present.

> "You are an expert peer-reviewer for the NeurIPS conference. You are provided with a PDF of a research paper from the 2024 Main Conference Track.
>
> This specific paper DOES NOT have a dedicated "Limitations" section.
>
> Your task is to carefully read the entire paper and extract the exact quotes of any limitations ONLY IF the authors explicitly identify them as limitations using specific language.
>
> ABSOLUTE REQUIREMENT: The authors MUST explicitly frame the sentence as a limitation or an ongoing challenge. You must only extract statements that contain clear identifying phrases such as:
>
> - "A limitation is..."
> - "The paper is limited because..."
> - "Our research is limited because..."
> - "One major limitation..."
> - "A weakness of our approach is..."
> - "Our method is limited by..."
> - "[Metrics/Methods/Features] are limited..."
> - "limitations of [concept/method]..."
> - "...remains an important challenge."
>
> If the authors do not explicitly call it a limitation or weakness using phrasing similar to the above, DO NOT EXTRACT IT.
>
> IMPORTANT: Sometimes, authors will describe a limitation and then refer to it in the next sentence (e.g., "This limitation of our method... so is a direction for future research."). In these cases, you MUST extract the sentence that contains the substance of the limitation, which is usually the sentence immediately BEFORE the one containing the reference. Do not extract the sentence that just refers to "this limitation" and points to future work.
>
> Please do NOT include:

- Sentences that only refer to a limitation described previously (e.g., "This limitation suggests future work..."). Extract the actual limitation text instead.
- Statements of scope (e.g., "is beyond the scope of this paper").
- Explanations of experimental design choices or omitted metrics (e.g., "standard deviations are not reported because it would be too computationally expensive").
- Factual reports of model size, parameter counts, or training time (e.g., "resulting in a 40% increase in parameter count").
- Discussions about hyperparameter tuning or sub-optimal parameters.
- Future work suggestions. Do NOT extract sentences that talk about "future research", "future work", or what "remains to be explored", even if they mention a current challenge.
- Background limitations (limitations of prior work by other authors).

CRITICAL INSTRUCTION: Do not infer, guess, or hallucinate limitations. If the authors are simply reporting their results, explaining a trade-off, defining their scope, or explaining why they didn't run a certain test, DO NOT EXTRACT IT. When in doubt, leave the list empty.

For each limitation, extract the exact textual statement verbatim from the paper. Do not paraphrase.

You must also extract the exact name of the main parent section where you found each limitation. Strip any leading numbers or letters from the section name (e.g., return "Conclusion" instead of "4 Conclusion"). If it is in a subsection, return ONLY the name of the main parent section (e.g., return "Appendix" instead of "A.7 Acceleration methods").

You must extract the exact, full title of the paper directly from the first page of the PDF.

Return the output in the following format:

```
{
    "paper_title": "Exact␣title␣extracted␣from
        ␣the␣PDF",
    "has_limitations_section": false,
    "limitations": [
        {
            "limitation_id": 0,
            "limitation_text": "Exact␣quote␣
                from␣the␣paper",
            "section_found": "Name␣of␣the␣main
                ␣section,␣without␣numbers"
        }
    ]
}
```

If no limitations are acknowledged anywhere in the paper, leave the "limitations" list empty, but still include "has_limitations_section": false.

You MUST return only the valid JSON object with no additional text, explanations, or formatting."