Context-Aware and Controllable Academic Paper Revision via Human-AI Collaboration

Anonymous ACL submission

Abstract

Despite the growing adoption of large language models (LLMs) in academic workflows, their capabilities remain limited to support high-quality scientific writing. Most existing systems are designed for general-purpose scientific text generation and fail to meet the sophisticated demands of research communication beyond surface-level polishing, such as conceptual coherence across sections. Furthermore, academic writing is inherently iterative and revision-driven, a process not well supported by direct prompting-based paradigms. To address these scenarios, we propose a human-AI collaboration framework for academic paper revision centered on criteria-guided intent alignment and context-aware modeling. To validate the framework, we curate a dataset of 7,000 research papers from top-tier venues annotated with 140,000 instruction–response pairs that reflect realistic, section-level scientific revisions. We instantiate the framework in `PaperCursor`, the first suite of open-source LLMs (1.5B to 14B parameters) for context-aware, instruction-guided writing assistance. Extensive experiments validate that `PaperCursor` significantly outperforms same-scale baselines and approaches the quality of proprietary systems. Both automated preference assessments and human evaluations confirm the effectiveness of `PaperCursor` in improving scientific drafts.

1 Introduction

Large language models (LLMs) are increasingly embedded in cognitively demanding workflows, particularly in scientific domains such as hypothesis generation (Zhou et al., 2025), proposal writing (Gottweis et al., 2025), and literature review (Asai et al., 2024). While some applications focus on generating scientific content from scratch (Weng et al., 2025; Yamada et al., 2025; Wang et al., 2025) (ethical concerns discussed in Appendix 5), academicians increasingly leverage LLMs as assistants for refining their academic paper drafts. In this setting, users request specific improvements (such as enhancing clarity or strengthening the motivation), and the model suggests revisions grounded in the original draft, as illustrated in Figure 1 (a). This human-AI collaborative revision process (Khalifa and Albadawy, 2024; Naddaf, 2025) helps preserve scientific originality while improving the clarity, coherence, and overall presentation of research ideas.

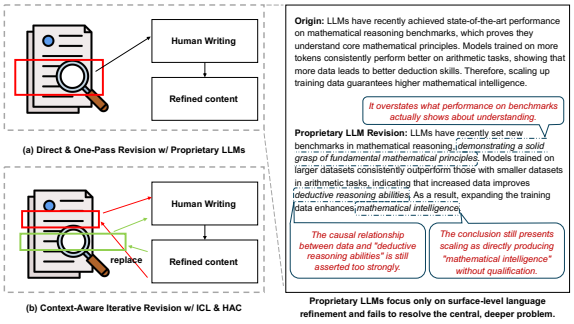


Figure 1: (Left) Overview of the academic paper revision workflow comparing proprietary LLMs and our method. (Right) An example of a poor revision generated by a proprietary LLM. A detailed case study of our model `PaperCursor` is provided in Table 4.

The prevailing mode of LLM-assisted paper revision involves prompting proprietary models such as GPT-4o (Hurst et al., 2024) through web interfaces. This approach is at best effective for surface-level editing, and faces two key limitations in the context of scientific writing. First, **general-purpose LLMs often lack explicit understanding about the deeper structure and argumentative rigor required for academic papers**. As shown in Figure 1, when prompted to revise an unclear motivation paragraph, GPT-4o improved the fluency of the text but failed to address the missing rationale, leaving the revised version equally unpersuasive or at times, overly belaboured in its points. Second, **scientific writing is inherently an iterative process** (Montgomery, 2017; Schimel, 2012) rather than generating from scratch. Authors revise their work over multiple rounds of drafting, incorporating feedback and improving content in a context-sensitive manner. Current LLM workflows do not naturally support this iterative nature. They treat each prompt in isolation and lack mechanisms to

Table 1: Comparison of existing full-paper AI generation systems across key dimensions: common quality issues, use of In-Context Learning (ICL), specific writing tasks, controllability, inclusion of Human–AI Collaborative (HAC) mechanisms, and whether the system generates content from scratch. Controllability is defined as the system's ability to adapt to user intent, provide fine-grained control over content generation, and support dynamic interaction during the writing process. Human–AI Collaborative (HAC) mechanisms involve iterative processes where human modifications to generated content are fed back into the model's *learning* or *generation*, enabling ongoing collaboration beyond one-off outputs without integrated feedback.

AI Generation System	ICL	Ctrl.	HAC	Quality Issues	Task
PaperRobot (Wang et al., 2019)	✔	✔	✔	Not LLM based, bad QA quality	Draft generation
August et al. (2022)	✗	✗	✗	Only definition	Scientific definition generation
STORM (Shao et al., 2024)	✔	✗	✗	Biased & Factual Hallucination	Article writing from scratch
CO-STORM (Jiang et al., 2024)	✔	✗	✔	Lack of Consistency	Information-seeking assistance system
CycleResearcher (Weng et al., 2025)	✗	✗	✗	Reward Hacking & Outdated	Paper generation and rating cycle
Ifargan et.al. (Ifargan et al., 2024)	✗	✔	✔	From scratch	Automatic stepwise research
Agent Lab (Schmidgall et al., 2025)	✗	✗	✗	Structure Rigidity	Report, experimentation, and writing
AI Scientist (Lu et al., 2024)	✗	✗	✗	No control idea	Review and Idea
Wang et al. (2025)	✗	✔	✔	From scratch	Survey
`PaperCursor`	✔	✔	✔	*Controllable revision*	Paper Revision

track changes or maintain context across revision cycles. This limits their effectiveness in real-world scientific authoring scenarios.

Human-AI Collaborative (HAC) paper revision demands an instruction-driven, fine-grained process that aligns closely with established norms of scientific writing. These norms are often formalized in detailed guidelines issued by top-tier conferences, such as the ICLR reviewer guide (ICLR, 2025). While recent studies have explored related tasks including end-to-end article generation (Weng et al., 2025), idea generation (Ghafarollahi and Buehler, 2024), automated paper evaluation (Zhu et al., 2025), and grammar correction (van Zeeland, 2023), they do not explicitly model the rationale-driven, feedback-centric nature of scientific revision. Specifically, they are uncontrollable in three ways: in-context following, user instructions, and generating output that conforms to the writing criteria. As a result, their generated content often suffers from issues in clarity, coherence, or alignment with scholarly standards, as summarized in Table 1.

To make LLMs more effective and accessible for scientific writing support, we propose a framework in which human authors retain creative control by generating ideas and drafting content, while LLMs act as targeted assistants that provide context-aware revisions. Our goal is to assist authors in producing improved writing, turning the writing process into a minimal overhead task. In Section 2, we detail the high-level principles and methodological design, emphasizing post-training for enhanced in-context learning (ICL) to enable controllable, iterative revisions via human-AI collaboration. On the data collection protocol, to simulate real-world scientific revision needs, each revision is guided by one of 20 section-level criteria informed by authoritative writing guides (Widom, 2006) and expert revisions contributed by experienced AI researchers (see Table 2). These pairs cover a diverse range of instruction-driven, section-specific edits grounded in realistic revision scenarios. Furthermore, we provide an evaluation framework to measure the controllability and effectiveness using length-controlled LLM-as-a-Judge, allowing future developments to be benchmarked and justified.

As an implementation (Section 3), we curate `ReviseQA`, a comprehensive dataset comprising 7,000 research papers from top-tier venues, annotated with 140,000 high-quality instruction–revision pairs. Building on `ReviseQA`, we introduce `PaperCursor`, the first family of **open-source** LLMs designed to support human-in-the-loop scientific writing, ranging from 1.5B to 14B parameters. Inspired by the modularity and interactivity of modern code editors (Cursor, 2024), `PaperCursor` allows users to revise specific sections of a paper through explicit instructions tied to writing goals (e.g., improving clarity, tightening motivation). By training on the criteria-guided `ReviseQA` dataset, `PaperCursor` internalizes both the structural expectations and rhetorical strategies characteristic of top-tier research writing. This enables the model to produce revisions that uphold academic rigor while remaining aligned with real-world authoring workflows.

We validate the effectiveness of `PaperCursor`

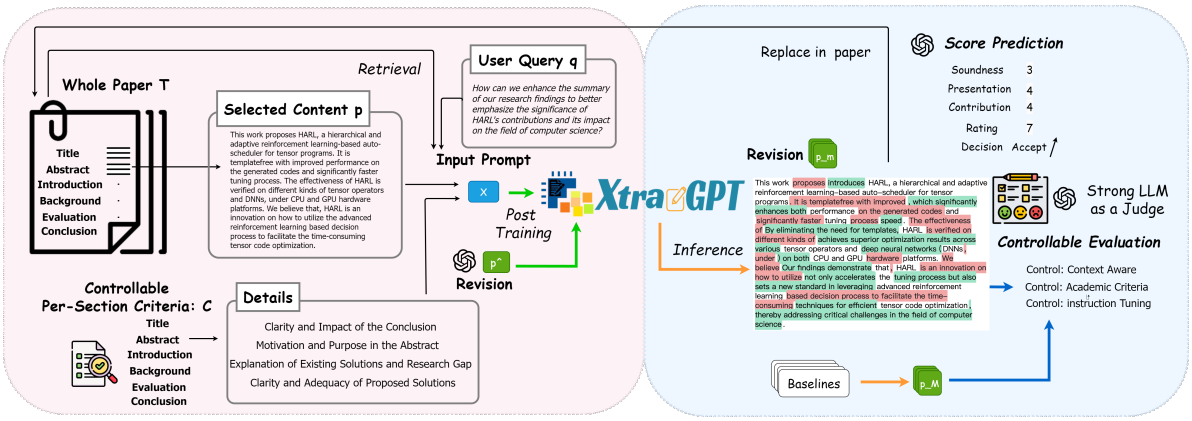


Figure 2: Overview. The post-training pipelines enable controllable, section-level, fine-grained paper revision.

with both quantitative and qualitative evaluations. LLM-as-a-Judge results using length-controlled win rates show our revisions are consistently preferred over original drafts. Model comparisons reveal that `PaperCursor` outperforms baseline models, with the 7B variant matching GPT-4o-mini and the 14B variant surpassing it. Human evaluations confirm that `PaperCursor` produces rationale-aligned improvements that users are willing to adopt; furthermore, the significant rise of rating predictions implies that incorporating these revisions leads to measurable gains in overall paper quality, with the predicted overall rating increased 0.65±0.15 from 6.08 to 6.73 (p<0.001).

2 Methodology

To enable LLMs to function as effective collaborators in academic writing, we propose a post-training framework designed to address the core challenges of context-awareness and controllability (in Figure 2). This section outlines the problem formulation, the guiding principles of our framework, and the corresponding training objective.

2.1 Problem Formulation

The task of controllable academic paper revision can be formally defined as follows. Given a full paper draft T, a specific paragraph $p \subset T$ to be revised, and a natural language instruction q from the author detailing the desired improvement, the goal is to generate a revised paragraph $\hat{p}$ that not only satisfies the instruction q but also maintains coherence with the global context T. The model, parameterized by θ, must learn the mapping:

$$\hat{p} = \text{Model}_\theta(p, q, T)$$

The key challenge lies in training Model_θ to effectively use the rich, structured information within T and q to produce high-quality, targeted revisions.

2.2 Guiding Principles

Our framework is built upon **Human-AI Collaboration (HAC)** as a foundational design philosophy *rather than a standalone technical contribution*. HAC represents our conceptual stance (more in Section 5): instead of pursuing full automation, which risks stifling originality, we cast the LLM as an assistant that augments authorial capabilities. In this paradigm, the human author directs the intellectual core (ideas, arguments, and initial drafts) while AI executes targeted, context-aware refinements based on specific instructions. This preserves authorial control and turns revision into an efficient, structured content. The philosophy is realized through two fundamental technical principles designed to make AI a more effective collaborator:

Criteria-Guided Intent Alignment. Author instructions are often high-level, possibly muddled, and tied to established norms of academic writing (e.g., "strengthen the contribution," "clarify the methodology"). To make such instructions actionable for an LLM, our second principle is that **training data must be structured around a predefined set of academic writing criteria** $\mathcal{C}$. These criteria, derived from authoritative writing guides and reviewer guidelines, serve as a bridge between abstract authorial intent and concrete textual modifications. By training on instruction-revision pairs that are explicitly linked to these criteria (e.g., q is an instantiation of a criterion $c \in \mathcal{C}$), the model learns to associate specific types of requests with corresponding revision strategies. This ensures *controllability* by aligning the model's behavior with the structured, goal-oriented nature of writing.

Context-Aware Modeling. To perform meaningful revisions, a model must understand a para-

Table 2: Section-level revision criteria covering six key components of scientific papers.

Aspect	Comments
Title	Consistency and Alignment of Title with Content Conciseness and Clarity of Title
Abstract	Clarity and Impact of the Conclusion Motivation and Purpose in the Abstract Explanation of Existing Solutions and Research Gap Clarity and Adequacy of Proposed Solutions
Introduction	Strength and Clarity of Motivation in the Introduction Review of Existing Approaches in Introduction Audience Alignment and Appropriateness Clarity and Visibility of Contributions Clarity and Specificity of Problem Definition Integration of State-of-the-Art in Problem Framing
Background	Contextual Relevance and Clarity of Background Coverage of Key Preliminary Concepts Clarity and Consistency of Terminology
Evaluation	Experimental Setup Clarity and Reproducibility Depth and Clarity of Figures and Tables Analysis Experimental Support for Main Innovations
Conclusion	Broader Impact and Future Directions Clarity and Impact of Key Innovations and Findings

graph's function within the entire document. A requested change to the 'Motivation' section in the introduction requires different considerations than a change to an 'Analysis' paragraph in the evaluation section. Our framework's principle is that **the full document context T must be an explicit input to the model during training and inference**. This forces the model to learn representations that are conditioned on the global narrative, structure, and terminology of the paper, enabling it to generate revisions that are contextually consistent and coherent. This principle directly facilitates *in-context learning* by providing the necessary scope for the model to ground its revisions.

2.3 Criteria-Guided Revision Data Curation

Training effective models for scientific writing revision requires more than surface-level language correction; it necessitates a deep understanding of what constitutes a high-quality research paper. To simulate real-world revision needs and guide models toward meaningful improvements, we define a set of 20 section-level revision criteria, denoted by $\mathcal{C}$ (see Table 2), spanning six core components of scientific papers: title, abstract, introduction, background, evaluation, and conclusion. These criteria are grounded in authoritative scientific writing guidelines (Widom, 2006) and further revised through expert revisions contributed by experienced AI researchers.

Using these curated criteria, the instruction–revision pairs could target specific improvements in selected paragraphs, enabling LLMs to learn revision strategies aligned with established standards of scientific communication.

2.4 Controllable Post-Training

To operationalize these principles, we formulate a Controllable Post-Training (CPT) objective. We create a dataset $\mathcal{D}_{CPT}$ where each instance is a tuple $(q, T, p, \hat{p})$, curated according to the principles above. The training objective is to maximize the conditional log-likelihood of the target revision $\hat{p}$ given the original paragraph p, the instruction q, and the full paper context T:

$$\mathcal{L}_{\text{CPT}}(\theta) = -\mathbb{E}_{(q,T,p,\hat{p})\sim\mathcal{D}_{CPT}} \left[\log P_\theta(\hat{p} \mid q, T, p)\right]$$

From the perspective of generative model alignment via high-quality demonstrations, we can observe the goal of optimization P_θ as $P_\theta(\hat{p} \mid q, T, p) = \sum_c P_\theta(\hat{p} \mid c, q, T, p) P_\theta(c \mid q, T, p)$. Since the instruction-revision pairs $(q, \hat{p})$ in our dataset are generated based on specific criteria $c \in \mathcal{C}$, this objective implicitly trains the model to approximate the distribution $P(\hat{p} \mid q, T, p, c)$. This alignment with structured criteria is key to improving the model's controllability, its ability to follow instructions q, comply with section-level criteria $\mathcal{C}$, and maintain contextual consistency with T.

2.5 Evaluation Protocol

Previous work (Tan et al., 2025) has used *surface-level* metrics such as ROUGE to evaluate full-text generation alignment in AI-assisted research. However, unlike tasks with a single ground truth, academic revision is inherently subjective. Such metrics only assess lexical overlap and fail to *capture instruction intent adherence, controllable revision behavior, or overall, the semantic quality of improvements*. Given **the lack of established benchmarks** for it, we adopt preference-based evaluation, specifically the **Length-Controlled Win Rate (LC-Win Rate)** (Dubois et al., 2024), which allows for comparative evaluation while accounting for response length bias (Appendix G). Judgments could be provided by an automatic evaluator, in our case, `alpaca_eval_gpt4_turbo_fn` (Zheng et al., 2023), which achieves 68.1% agreement with human evaluations according to (Tatsu-lab, 2023). We adapt the scoring prompts (Figure 7) and comparison prompts (Figure 6) to focus on controllability and alignment with instruction intent.

Why use LLM-as-a-Judge for controllable paper revision? For subjective sequence-level tasks such as paragraph rewriting, LLM-based evaluation

provides consistent and scalable feedback (e.g., on controllability). This practice has proven effective in the development of models such as InstructGPT and ChatGPT. Prior work has shown that LLM judges align well with human reviewers in research settings (Lu et al., 2024). Compared to manual annotation, this approach offers faster feedback while preserving evaluation quality.

Length-Controlled Win Rate Formulation. Let m denote the target model and M denote a baseline model. Let θ be the raw prediction score, and let z_m and z_M denote their respective outputs given input x. The length-controlled win rate is computed as:

$$q_{\theta,\phi,\psi}(y = m \mid z_m, z_M, x) := \text{logistic}\left(\underbrace{\theta_m - \theta_M}_{\text{model term}} + \underbrace{\phi_{M,b} \cdot \tanh\left(\frac{\text{len}(z_m) - \text{len}(z_M)}{\text{std}(\text{len}(z_m) - \text{len}(z_M))}\right)}_{\text{length term}}\right)$$

The instruction difficulty term in this work is omitted to focus solely on the effectiveness of the revisions. The final LC win rate is computed as:

$$\text{winrate}^{LC}(m, M) = 100 \cdot \mathbb{E}_x[q_{\theta,\phi,\psi}(y = m | z_m, z_M, x)]$$

This formulation adjusts for output length variation, ensuring that model comparisons reflect actual quality improvements rather than superficial verbosity.

3 Instantiation

To validate the framework proposed in Section 2, we conducted a large-scale study. This section details the specific implementation of our data curation process for model training, designed to serve as a concrete testbed for our methodology.

ReviseQA Following the principles outlined in our framework, we created `ReviseQA`, the first large-scale dataset designed for controllable, context-aware paper revision. The curation process itself is a simulation of our envisioned **Human-AI Collaboration workflow**, designed to capture the essence of iterative, instruction-driven refinement.

Source Data Acquisition and Context Modeling. To ensure the data reflects high-quality, contemporary scientific writing, we begin by collecting all research papers submitted to ICLR 2024 via the OpenReview API. Faced by practical constraints on time and computation, this selection was motivated by ICLR's high paper quality and data transparency, and the timeliness of the 2024 dataset, as its guidelines had been updated and 2025 data was not yet released during our project. Review statistics, including soundness, presentation, contribution, and overall ratings, are shown in Figure 19, and the distribution of paper lengths is presented in Figure 8. Among the submissions, 64.71% received reviewer replies, with 82.4% of those discussions leading to a final decision. Based on the parsed and filtered PDF files, the overall acceptance rate was 36.3%.

After excluding 40 out of 7,040 excessively long papers, we retained 7,000 valid submissions. Each PDF was converted into structured markdown using NOUGAT (Blecher et al., 2023), a deep learning-based academic parser that outperforms prior rule-based approaches, as demonstrated by Li et al. (2024d). The resulting set of tokenizable paper texts is denoted as $\mathcal{T}$.

To ensure length consistency and content relevance, we applied a post-processing step to remove non-essential sections, such as acknowledgments and references, retaining only the main body of the paper. This allowed $\mathcal{T}$ to remain within the maximum token limit of 16,384 tokens, making it suitable for downstream modeling. To adhere to our principle of *Context-Aware Modeling*, we retained the main body of each paper as the full context T, providing the AI with the same global view a human collaborator would have.

Simulating Collaborative Interaction via Criteria-Guided Curation. Using these curated criteria in Table 2, we generate instruction–revision pairs that target specific improvements in selected paragraphs, enabling LLMs to learn revision strategies aligned with established standards of scientific communication. For each article $T \in \mathcal{T}$, we sample user-identifiable paragraphs from the aforementioned six core sections, and generate 20 criterion-guided questions using carefully constructed prompts. For each sampled paragraph $p \in T$, we first generate an instruction query q based on a selected criterion $c \in \mathcal{C}$. Given the full paper context T, with limited funds, we then use GPT-4o-mini, a model with a reported hallucination rate of only 1.7% (Hong et al., 2024), to generate the corresponding revised paragraph, denoted as $\hat{p}$. We selected GPT-4o-mini for data generation instead of more recent state-of-the-art LLMs (e.g., DeepSeek R1 (DeepSeek-AI, 2025)) because our task does not require complex reasoning, multi-step planning, or deep domain

knowledge. Rather, it emphasizes long-context understanding and localized revision, areas in which GPT-4o-mini performs reliably. As validated by human annotators (see Table 15), GPT-4o-mini produces coherent and high-quality paragraph-level revisions suitable for this task. The full set of criteria is illustrated in Figures 9 through 14, and the prompt design is detailed in Figure 5.

These pairs form the foundation of our ReviseQA dataset, which enable models to learn principled, context-aware revisions grounded in realistic scientific writing goals. In total, this process yields 140,000 high-quality instruction–response pairs. To facilitate rigorous evaluation, we construct a held-out QA benchmark by randomly sampling 5% of the papers and their corresponding instruction-revision pairs, totaling 350 papers and 7,000 QA pairs. Our benchmark supports model comparison using length-controlled win rate evaluation (Dubois et al., 2024), with PaperCursor serving as the anchor model.

We formally define the dataset used for Contollable Post-Training (CPT) as $\mathcal{D}_{CPT}$, where each data instance $D \in \mathcal{D}_{CPT}$ is in the form of $D \in \mathcal{D}_{CPT} = \{(q, T, p, \hat{p})\}$.

Data Quality Validation. To ensure the reliability of ReviseQA, we implemented strict quality controls, culminating in a human evaluation. Three Ph.D. students in computer science evaluated a random sample of instruction–revision pairs on four key aspects: instruction following, criteria alignment, contextual relevance, and overall revision acceptance. As shown in Table 15 in Appendix D, the high agreement and acceptance rates confirm that ReviseQA is a valid and high-quality resource for training and evaluation.

4 Experiment

We then conduct experiments on ReviseQA to answer the following research questions:

- **Effectiveness** (Section 4.2): How does PaperCursor perform in comparison to representative LLM baselines?
- **Adaptability** (Section 4.2): Can PaperCursor be effectively integrated with different architectures and sizes?
- **Human Preference** (Section 4.3): How are the revisions by PaperCursor perceived in terms of quality and usefulness by human evaluators?
- **Real-World Applicability** (Section 4.4): How well does PaperCursor support practical human–AI collaborative paper revision workflows?

4.1 Experimental Setup

Benchmarking. We benchmark representative LLMs on ReviseQA to evaluate controllable paper revision capabilities. During preliminary analysis, we identified several common issues with LLM-based revisions: overuse of generic GPT-style language (e.g., "comprehensive") that artificially inflates the perceived impact of the paper, superficial edits that do not address the core revision intent, and a tendency to generate unnecessarily long revised segments. To mitigate these issues, we carefully designed our generation prompt (Figure 5) to encourage focused and contextually grounded improvements. Additionally, we introduced evaluation-specific prompts (Figures 6 and 7) that guide models and human judges to assess revisions with an emphasis on clarity, conciseness, and content relevance.

PaperCursor Training Setup. PaperCursor models are designed of different sizes: 1.5B (based on Qwen-2.5-1.5B-Instruct), 3.8B (based on phi3.5-3.8b), 7B (based on Qwen-2.5-7B-Instruct), and 14B (based on phi4-14b). Training was conducted using 4 NVIDIA H100 GPUs, each with 80 GB of memory. For inference on ReviseQA, we employed the vLLM framework (Kwon et al., 2023) on a system with a single NVIDIA A100 GPU (80 GB). The computing environment was configured with CUDA 12.2 and cuDNN 9.1 to ensure optimal deep learning performance.

4.2 Comparison with Paper Revision Models

As shown in the length-controlled win rate results in Table 3, **PaperCursor-7B revisions are consistently preferred over same-scale (7B/8B) open-source models**. Moreover, PaperCursor-7B outperforms GPT-4o-mini on several sections, including the abstract, evaluation, and conclusion. When scaling the model to 14B parameters, the larger PaperCursor variant ranks higher on the leaderboard than GPT-4o-mini, demonstrating strong performance at scale. A case study in Table 4 further highlights PaperCursor-7B's practical applicability over GPT-4o's by showcasing nuanced improvements beyond numerical metrics.

To evaluate the generalizability of PaperCursor framework, and as an implicit ablation study on

Table 3: Length-controlled (LC) win rates of various models against the 7B version of `PaperCursor` (used as the *anchor*) across different evaluation categories. Models above the grey anchor bar surpass `PaperCursor`; those below under-perform. Evaluations are conducted using a modified `alpaca_eval_gpt4_turbo_fn` judge with the prompt shown in Figure 6.

Models	**Title**	**Abstract**	**Introduction**	**Background**	**Evaluation**	**Conclusion**	**Overall↑**
QwQ-32B	46.58	85.34	81.99	83.82	82.64	95.69	80.86
DeepSeek-v3-671B	56.42	65.71	68.32	74.12	72.11	64.83	67.70
Qwen2.5-72B-Instruct	50.35	56.78	64.75	67.13	64.59	60.34	61.61
PaperCursor-14B	55.29	59.43	50.90	59.43	57.87	52.11	55.49
GPT-4o-Mini	48.80	47.43	55.73	66.07	45.67	39.03	51.75
`PaperCursor-7B`; base: `Qwen-2.5-7B-Instruct`(*anchor*↑)							
Qwen2.5-7B-Instruct	39.93	45.14	45.64	39.28	33.87	31.17	40.80
Phi-4	4.57	44.78	50.90	52.54	26.33	42.01	40.71
Qwen-QWQ-32B-Preview	37.83	34.57	32.13	40.58	30.04	32.91	34.22
Llama-3.1-8B-Instruct	34.78	30.64	35.31	41.60	40.29	18.36	33.51
DeepSeek-R1-Distill-Llama-8B	37.98	15.19	10.87	9.64	9.73	9.38	13.94
Qwen2.5-1.5B-Instruct	36.07	30.87	25.80	21.34	24.18	24.27	26.80
GPT-3.5-Turbo	25.73	23.99	21.52	23.16	30.97	17.39	24.24
Llama-3.2-3B-Instruct	19.93	6.45	9.35	3.80	8.26	4.64	8.73

Shared Content (from abstract): ... propose CVG framework based on VGAE and GIB for generalizable graphs. Experiments show superiority over baselines.
Q.: Summarize innovations, contributions, specific results (e.g., % improvements).

GPT-4o Response:	**PaperCursor Response:**
Excerpt: "Our framework is implemented based on [VGAE] ... optimized under [GIB] ... key innovations include: Constrained Variational Generation ... GIB Optimization ... theoretical analysis validates ..."	Excerpt: "... we propose a method that ... ensures ... aligned with the underlying task. ... experimental results demonstrate that [it] significantly outperforms ... across various complex and zero-shot IL tasks ..."
Weaknesses: No length control; simple paraphrase; generalized; single interaction; chatbot-like, without human-ai collaboration.	**Strengths:** Holistic processing (e.g., refers to experiment results); critical feedback; maintains flow of thoughrs.

Table 4: Condensed case study comparing GPT-4o and PaperCursor. Full version is provided in Appendix L.

data curation, we trained and compared models based on representative open-source LLM backbones of varying sizes, ranging from 1.5B to 14B parameters. The corresponding results are reported in Table 5. Across all configurations, `PaperCursor` consistently outperforms its respective base model. Base models rely on general-purpose capabilities from broad pre-training, whereas `PaperCursor` is fine-tuned on our specialized dataset, structured by academic writing criteria. The significant improvement demonstrates that **criteria-guided fine-tuning yields substantial gains over general abilities, validating the effectiveness of our approach.**

4.3 Qualitative Evaluation

Human Perception of Revision Quality. Beyond the length-controlled win rate results presented in Table 3 and Table 5, we further assess the effectiveness of revisions through human evaluation by measuring users' willingness to adopt the generated revision in place of the original contents.

Table 6: Effectiveness of `PaperCursor` as validated by human ratings on a 1–5 scale.

`PaperCursor`	**Judge 1**	**Judge 2**	**Judge 3**	**Aggregated**
Instruction Following	3.25	3.99	4.09	3.78
Criteria Alignment	3.34	3.70	4.09	3.71
In-Context Reference	2.80	3.40	4.06	3.42
Revision Acceptance	2.46	3.23	4.01	3.23

Following a similar evaluation setup to our dataset quality validation (Section 3), we engaged three evaluators with research backgrounds and provided them with detailed criteria for assessing model performance across four key dimensions. These criteria are visualized in Figure 15. Table 6 reports the human-assigned quality ratings of `PaperCursor` revisions, demonstrating that the outputs are consistently perceived as high-quality and aligned with human expectations. Furthermore, Table 4 shows a concrete comparison of `PaperCursor`-revised paper excerpt and the LLM-revised paper excerpt.

4.4 Reliability of HAC Revision

To assess the effectiveness of our paper improvement tool, we employed AI-SCIENTIST (Lu et al., 2024), an open-source system developed by Sakana AI. AI-SCIENTIST includes a functionality that simulates the peer review process and provides **paper-level** evaluations. Notably, the system's capability is underpinned by the success in generating a fully AI-authored paper that passed peer review at a top-tier conference workshop (Yamada et al., 2025). Before applying the system to our own data, we first establish the reliability of AI-SCIENTIST as an evaluator of scientific writing quality.

To this end, we validated AI-SCIENTIST by ap-

Table 5: Length-controlled (LC) win rates of various models against `PaperCursor` (*anchor*) across evaluation categories. Models are ranked in descending order based on their weighted LC win rates. A modified version of `alpaca_eval_gpt4_turbo_fn` was employed as judge (see Prompt 6).

Models	**Title**	**Abstract**	**Introduction**	**Background**	**Evaluation**	**Conclusion**	**Overall↑**
			`PaperCursor`-7B (*anchor*↑)				
Qwen2.5-7B-Instruct	39.93	45.14	45.64	39.28	33.87	31.17	40.80
			`PaperCursor`-14B(*anchor*↑)				
Phi-4 (14B)	5.09	40.11	39.63	50.00	24.27	39.08	35.47
			`PaperCursor`-3.8B (*anchor*↑)				
Phi-3.5-mini-instruct (3.8B)	17.52	43.74	31.64	34.41	25.85	58.24	34.86
			`PaperCursor`-1.5B(*anchor*↑)				
Qwen2.5-1.5B-Instruct	16.83	10.70	10.34	5.41	4.39	15.81	9.98
			`PaperCursor`-3B (*anchor*↑)				
Llama-3.2-3B-Instruct	0.14	0.57	0.56	0.38	0.26	1.89	0.58

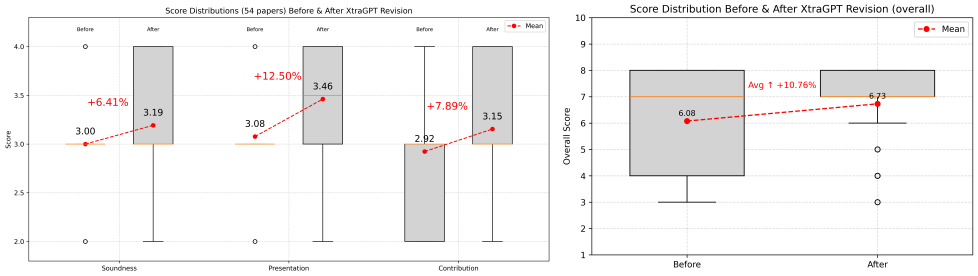


Figure 3: Paper quality scores and overall ratings from o1-based AI-SCIENTIST, before and after `PaperCursor` revision. **Left:** Evaluation of revision quality. On average, contribution scores increased by 7.89%, presentation by 12.50%, and soundness by 6.41%. **Right:** Distribution of overall ratings on a 1–10 scale before and after revision. Average rating increased by 10.76%.

Table 7: Performance on predicting decisions.

Metric	**Value (%)**
Accuracy	70.57
Precision (Accept)	57.14
Recall (Accept)	80.00
F_1 Score (Accept)	66.64

plying it to a sample of 54 ICLR 2024 submissions with known ground-truth acceptance decisions. The system was tasked with predicting the acceptance outcome for each paper. Results indicate that AI-SCIENTIST achieves a reasonable level of accuracy and reliability in approximating peer review judgments, thereby justifying its use as an automated evaluator in our study. A summary of these results is provided in Table 7.

Following the validation of AI-SCIENTIST's reliability, we employed it to evaluate the full-paper quality improvements made by `PaperCursor`. Specifically, AI-SCIENTIST was used to assign scores on three core dimensions from the review rubric: *contribution*, *presentation*, and *soundness*, each rated on a 1–4 scale, and *overall rating* on a 1–10 scale. For each of the 54 papers in our evaluation set, we obtained these scores both before and after revision by `PaperCursor` (repeat 3 times for t-test). As shown in the left of Figure 3, `PaperCursor` yielded consistent gains across all three dimensions. For evaluation of revision quality, contribution (1-4 scale) raises 0.23±0.10 from 2.92 to 3.15 ($p<0.001$). Presentation (1-4 scale) raises 0.38±0.10 from 3.08 to 3.46 ($p<0.001$). Soundness (1-4 scale) raises 0.19±0.08 from 3.00 to 3.19 ($p=0.004$). In addition to these criteria-specific improvements, the overall score from 1 to 10 which reflects the holistic recommendation, also improved, increasing 0.65±0.15 from 6.08 to 6.73 ($p<0.001$) as shown in the right of Figure 3. These suggest that `PaperCursor`'s revisions lead to measurable improvements in dimensions that are central to peer review.

5 Conclusion

This paper presents a **methodological shift** for applying LLMs to academic writing, moving beyond generic text generation towards a controllable, collaborative revision process. Our core contribution is a framework that leverages full-document context and explicit, **criteria-guided** instructions to enable fine-grained, **context-aware** paper improvements. By formulating a controllable post-training objective based on this principle, we successfully align model outputs with the rigorous demands of scholarly communication. The empirical success of this approach validates that a focus on controllable, iterative refinement, rather than autonomous generation, is a more trustworthy paradigm for **human-AI collaboration** in academia (discussed below).

Discussion and Broader Implications

Limitations

We discuss several limitations in the current work. First, the `ReviseQA` dataset, while comprehensive for its design goals, is constructed from papers in top-tier venues of the AI/ML domain and leverages LLMs for generating instruction–revision pairs. This introduces potential limitations in terms of domain specificity and generator bias. The learned revision strategies may be heavily tailored to the writing style and norms prevalent in these venues, potentially limiting generalizability to other scientific disciplines.

Second, while `PaperCursor` demonstrates strong performance on instruction-guided paragraph revision, it is an LLM-based system and shares inherent limitations of this technology. The model's ability to maintain long-term context or internal state across multiple, complex iterative revision cycles on a full paper remains an area for further exploration.

Third, evaluating the true impact of fine-grained revisions on overall paper quality is inherently challenging. Our reliance on LLM-as-a-Judge metrics (LC-Win Rate) and even the AI-SCIENTIST tool for full-paper scoring (Appendix O) faces limitations. LLM judges may not perfectly align with human expert reviewers on all aspects of scientific rigor and novelty. The difficulty of reliably scoring full papers with current AI tools highlights the needs for a definitive, automated measure of the impact of `PaperCursor`'s revisions.

Finally, as discussed in Section 5, the potential for over-reliance, introduction of bias from training data, and challenges in interpreting model suggestions are inherent risks associated with LLM-assisted writing tools. While our collaborative design aims to mitigate these, they represent limitations that must be carefully considered during deployment and use.

Potential Risks and Ethical Statement

This work does not advocate the use of LLM to replace human creativity or research ethics standards. Key ethical considerations include Authorship and originality (may blur authorial voice but can be mitigated by human control), bias in training data (ICLR papers, expert annotators), quality and accuracy (may be illusory in scientific contexts and require mandatory human validation), dependence and misuse (may degrade skills and polish flawed work), and transparency (limited interpretation of specific recommendations). Our human-computer collaborative design and open source approach aims to promote responsible and transparent use.

Broader Impact

The future AI-assisted academic research raises critical concerns. We analyze these concerns from the perspective of `PaperCursor`.

- **Potential for Human Researcher Passivity**: One widespread concern is the potential for over-reliance on AI, leading to diminished human effort, creativity, and critical thinking, as AI could handle various stages like idea generation, writing, and reviewing. Our PaperCursor framework, however, adheres to a human-AI collaboration paradigm where the human researcher retains agency and control. Authors are required to have a strong motivation to generate core ideas and initial drafts, reflecting their intellectual investment and desire for recognition. This collaborative process can be viewed as a positive *feedback loop*: the AI's assistance in refining the presentation of core ideas through revisions can, via psychological phenomena such as the *self-fulfilling prophecy*, *reinforce* the human author's motivation and drive for high-quality output. This fosters a virtuous cycle that encourages authors to be more active and engaged in producing and refining their work, rather than becoming passive.

- **Proliferation of Low-Quality Papers and Quantity Inflation**: AI, particularly in uncontrolled end-to-end generation scenarios, poses a risk of enabling the mass production of low-quality or superficially polished papers, potentially inflating publication numbers without commensurate scientific value. In the PaperCursor framework, the AI functions as an assistant specifically for improving existing drafts based on explicit instructions and established writing criteria (such as those informed by academic guides). The initial effort required from the human author to develop high-quality ideas and preliminary drafts is significant and remains a crucial, valuable step that underpins the current positive development of the research community. PaperCursor is designed to help authors present their

"already valuable" work more effectively and rigorously, implicitly discouraging the dissemination of poorly conceived work and supporting the critical refinement process that characterizes high-quality academic output, thereby helping the community manage article quality rather than promoting quantity over substance.

- **Misalignment with Human Values and Scientific Principles**: Concerns exist that AI might generate content that deviates from human researchers' values, core scientific principles, ethical considerations, or specific conference norms. PaperCursor's emphasis on controllability and instruction-following is designed to mitigate this risk. The model is trained and operates under constraints that aim to keep it closely aligned with the human author's core intent and the integrity of the original manuscript. In every collaborative interaction, the human author maintains overall judgment and intellectual control, ensuring that the final revised content reflects their will and adheres to academic standards and ethical guidelines, which are implicitly learned from the training data and explicitly guided by user instructions.

This work specifically targets the iterative process of paper revision, which is crucial for refining scientific communication, and aims to offer novel insights and tools to the community. We strongly advocate for increased attention to the ability of large models to adhere to core scientific principles and community standards, and emphasize that evaluation metrics for AI in academic assistance should move beyond traditional natural language processing scores to incorporate measures of adherence to these critical norms and principles. This consideration is vital for the responsible development and deployment of AI across all research assistance tasks.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Yiming Ai, Zhiwei He, Ziyin Zhang, Wenhong Zhu, Hongkun Hao, Kai Yu, Lingjun Chen, and Rui Wang. 2024. Is cognition and action consistent or not: Investigating large language model's personality. *arXiv preprint arXiv:2402.14679*.

Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. 2024. Litsearch: A retrieval benchmark for scientific literature search. *arXiv preprint arXiv:2407.18940*.

Barrett R Anderson, Jash Hemant Shah, and Max Kreminski. 2024. Homogenization effects of large language models on human creative ideation. In *Proceedings of the 16th Conference on Creativity & Cognition*, pages 413–425.

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, David Wadden, Matt Latzke, Minyang Tian, Pan Ji, Shengyan Liu, Hao Tong, Bohao Wu, Yanyu Xiong, Luke Zettlemoyer, and 6 others. 2024. Openscholar: Synthesizing scientific literature with retrieval-augmented lms. *Preprint*, arXiv:2411.14199.

Joshua Ashkinaze, Julia Mendelsohn, Li Qiwei, Ceren Budak, and Eric Gilbert. 2024. How ai ideas affect the creativity, diversity, and evolution of human ideas: Evidence from a large, dynamic experiment. *arXiv preprint arXiv:2401.13481*.

Tal August, Katharina Reinecke, and Noah A. Smith. 2022. Generating scientific definitions with controllable complexity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8298–8317, Dublin, Ireland. Association for Computational Linguistics.

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 2024. Researchagent: Iterative research idea generation over scientific literature with large language models. *arXiv preprint arXiv:2404.07738*.

Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. 2023. Nougat: Neural optical understanding for academic documents. *Preprint*, arXiv:2308.13418.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. 2024.

Art or artifice? large language models and the false promise of creativity. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–34.

Guhong Chen, Liyang Fan, Zihan Gong, Nan Xie, Zixuan Li, Ziqiang Liu, Chengming Li, Qiang Qu, Shiwen Ni, and Min Yang. 2024a. Agentcourt: Simulating court with adversarial evolvable lawyer agents. *arXiv preprint arXiv:2408.08089*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Xiuying Chen, Tairan Wang, Taicheng Guo, Kehan Guo, Juexiao Zhou, Haoyang Li, Mingchen Zhuge, Jürgen Schmidhuber, Xin Gao, and Xiangliang Zhang. 2024b. Scholarchemqa: Unveiling the power of language models in chemical research question answering. *arXiv preprint arXiv:2407.16931*.

CoWriter. 2025. Cowriter - your ai platform for speeding up creative writing.

Cursor. 2024. Cursor - the ai code editor.

Mike D'Arcy, Tom Hope, Larry Birnbaum, and Doug Downey. 2024. Marg: Multi-agent review generation for scientific papers. *arXiv preprint arXiv:2401.04259*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *Preprint*, arXiv:2404.04475.

Aaron Grattafiori et.al. 2024a. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

An Yang et.al. 2024b. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Yubin Ge, Neeraja Kirtane, Hao Peng, and Dilek Hakkani-Tür. 2025. Llms are vulnerable to malicious prompts disguised as scientific language. *Preprint*, arXiv:2501.14073.

Alireza Ghafarollahi and Markus J Buehler. 2024. Sciagents: Automating scientific discovery through multi-agent intelligent graph reasoning. *arXiv preprint arXiv:2409.05556*.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, and 1 others. 2025. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*.

Giwon Hong, Aryo Pradipta Gema, Rohit Saxena, Xiaotang Du, Ping Nie, Yu Zhao, Laura Perez-Beltrachini, Max Ryabinin, Xuanli He, Clémentine Fourrier, and Pasquale Minervini. 2024. The hallucinations leaderboard - an open effort to measure hallucinations in large language models. *CoRR*, abs/2404.05904.

Tiancheng Hu and Nigel Collier. 2024. Quantifying the persona effect in llm simulations. *arXiv preprint arXiv:2402.10811*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

ICLR. 2025. Iclr 2025 reviewer guide. https://iclr.cc/Conferences/2025/ReviewerGuide. Accessed: 2025-04-20.

Tal Ifargan, Lukas Hafner, Maor Kern, Ori Alcalay, and Roy Kishony. 2024. Autonomous llm-driven research from data to human-verifiable research papers. *Preprint*, arXiv:2404.17605.

Yucheng Jiang, Yijia Shao, Dekun Ma, Sina J. Semnani, and Monica S. Lam. 2024. Into the unknown unknowns: Engaged human learning through participation in language model agent conversations. *Preprint*, arXiv:2408.15232.

Hao Kang and Chenyan Xiong. 2024. Researcharena: Benchmarking llms' ability to collect and organize information as research agents. *arXiv preprint arXiv:2406.10291*.

Mohamed Khalifa and Mona Albadawy. 2024. Using artificial intelligence in academic writing and research: An essential productivity tool. *Computer Methods and Programs in Biomedicine Update*, 5:100145.

Chuyi Kong, Yaxin Fan, Xiang Wan, Feng Jiang, and Benyou Wang. 2024. PlatoLM: Teaching LLMs in multi-round dialogue via a user simulator. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7841–7863, Bangkok, Thailand. Association for Computational Linguistics.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2024. Benchmarking cognitive biases in large language models as evaluators. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 517–545.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

Jakub Lála, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G Rodriques, and Andrew D White. 2023. Paperqa: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*.

Bruce W Lee, Yeongheon Lee, and Hyunsoo Cho. 2024. Language models show stable value orientations across diverse role-plays. *arXiv preprint arXiv:2408.09049*.

Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, and 1 others. 2024a. Chain of ideas: Revolutionizing research via novel idea development with llm agents. *arXiv preprint arXiv:2410.13185*.

Sihang Li, Jin Huang, Jiaxi Zhuang, Yaorui Shi, Xiaochen Cai, Mingjun Xu, Xiang Wang, Linfeng Zhang, Guolin Ke, and Hengxing Cai. 2024b. Scilitllm: How to adapt llms for scientific literature understanding. *arXiv preprint arXiv:2408.15545*.

Yuan Li, Bingqiao Luo, Qian Wang, Nuo Chen, Xu Liu, and Bingsheng He. 2024c. CryptoTrade: A reflective LLM-based agent to guide zero-shot cryptocurrency trading. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1094–1106, Miami, Florida, USA. Association for Computational Linguistics.

Zichao Li, Aizier Abulaiti, Yaojie Lu, Xuanang Chen, Jia Zheng, Hongyu Lin, Xianpei Han, and Le Sun. 2024d. Readoc: A unified benchmark for realistic document structured extraction. *Preprint*, arXiv:2409.05137.

Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Yi Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Scott Smith, Yian Yin, and 1 others. 2024. Can large language models provide useful feedback on research papers? a large-scale empirical analysis. *NEJM AI*, 1(8):AIoa2400196.

Jiaju Lin, Haoran Zhao, Aochi Zhang, Yiting Wu, Huqiuyue Ping, and Qin Chen. 2023. Agentsims: An open-source sandbox for large language model evaluation. *arXiv preprint arXiv:2308.04026*.

Xinna Lin, Siqi Ma, Junjie Shan, Xiaojing Zhang, Shell Xu Hu, Tiannan Guo, Stan Z Li, and Kaicheng Yu. 2024. Biokgbench: A knowledge graph checking benchmark of ai agent for biomedical science. *arXiv preprint arXiv:2407.00466*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Yiren Liu, Si Chen, Haocong Cheng, Mengxia Yu, Xiao Ran, Andrew Mo, Yiliu Tang, and Yun Huang. 2024b. How ai processing delays foster creativity: Exploring research question co-creation with an llm-based agent. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–25.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The AI Scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, pages 1–11.

Scott L Montgomery. 2017. *The Chicago guide to communicating science*. University of Chicago Press.

Miryam Naddaf. 2025. How are researchers using ai? survey reveals pros and cons for science. *Nature*.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2022. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Vishakh Padmakumar and He He. 2024. Does writing with language models reduce content diversity? In *The Twelfth International Conference on Learning Representations*.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *In the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*, UIST '23, New York, NY, USA. Association for Computing Machinery.

Nikolay B Petrov, Gregory Serapio-García, and Jason Rentfrow. 2024. Limited ability of llms to simulate human psychological behaviours: a psychometric analysis. *arXiv preprint arXiv:2405.07248*.

Ori Press, Andreas Hochlehnert, Ameya Prabhu, Vishaal Udandarao, Ofir Press, and Matthias Bethge. 2024. Citeme: Can language models accurately cite scientific claims? *arXiv preprint arXiv:2407.12861*.

Joshua Schimel. 2012. *Writing science: how to write papers that get cited and proposals that get funded*. OUP USA.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using llm agents as research assistants. *Preprint*, arXiv:2501.04227.

Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Monica S. Lam. 2024. Assisting in writing wikipedia-like articles from scratch with large language models. *Preprint*, arXiv:2402.14207.

Wei Shen, Rui Zheng, Wenyu Zhan, Jun Zhao, Shihan Dou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Loose lips sink ships: Mitigating length bias in reinforcement learning from human feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2859–2873.

Zhengliang Shi, Shen Gao, Zhen Zhang, Xiuying Chen, Zhumin Chen, Pengjie Ren, and Zhaochun Ren. 2023. Towards a unified framework for reference retrieval and related work generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5785–5799, Singapore. Association for Computational Linguistics.

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

Xiaoshuai Song, Muxi Diao, Guanting Dong, Zhengyang Wang, Yujia Fu, Runqi Qiao, Zhexu Wang, Dayuan Fu, Huangxuan Wu, Bin Liang, and 1 others. 2024. Cs-bench: A comprehensive benchmark for large language models towards computer science mastery. *arXiv preprint arXiv:2406.08587*.

Weiwei Sun, Zhengliang Shi, Wu Jiu Long, Lingyong Yan, Xinyu Ma, Yiding Liu, Min Cao, Dawei Yin, and Zhaochun Ren. 2024. MAIR: A massive benchmark for evaluating instructed retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14044–14067, Miami, Florida, USA. Association for Computational Linguistics.

Kyle Swanson, Wesley Wu, Nash L Bulaong, John E Pak, and James Zou. 2024. The virtual lab: Ai agents design new sars-cov-2 nanobodies with experimental validation. *bioRxiv*, pages 2024–11.

Cheng Tan, Dongxin Lyu, Siyuan Li, Zhangyang Gao, Jingxuan Wei, Siqi Ma, Zicheng Liu, and Stan Z. Li. 2025. Peer review as a multi-turn and long-context dialogue with role-based interactions: Benchmarking large language models.

Shuo Tang, Xianghe Pang, Zexi Liu, Bohan Tang, Rui Ye, Xiaowen Dong, Yanfeng Wang, and Siheng Chen. 2024. Synthesizing post-training data for llms through multi-agent simulation. *arXiv preprint arXiv:2410.14251*.

Tatsu-lab. 2023. Alpacaeval: An automatic evaluator for instruction-following language models.

Qwen Team. 2024. Qwq: Reflect deeply on the boundaries of the unknown.

Hilde van Zeeland. 2023. Texgpt: Harness the power of chatgpt in overleaf.

Haoyu Wang, Yujia Fu, Zhu Zhang, Shuo Wang, Zirui Ren, Xiaorong Wang, Zhili Li, Chaoqun He, Bo An, Zhiyuan Liu, and Maosong Sun. 2025. Llm×mapreduce-v2: Entropy-driven convolutional test-time scaling for generating long-form articles from extremely long resources. *Preprint*, arXiv:2504.05732.

Qian Wang, Tianyu Wang, Qinbin Li, Jingsheng Liang, and Bingsheng He. 2024. Megaagent: A practical framework for autonomous cooperation in large-scale llm agent systems. *Preprint*, arXiv:2408.09955.

Qingyun Wang, Lifu Huang, Zhiying Jiang, Kevin Knight, Heng Ji, Mohit Bansal, and Yi Luan. 2019. PaperRobot: Incremental draft generation of scientific ideas. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1980–1991, Florence, Italy. Association for Computational Linguistics.

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. 2025. Cycleresearcher: Improving automated research via automated review. In *The Thirteenth International Conference on Learning Representations*.

Jennifer Widom. 2006. Tips for writing technical papers. `https://cs.stanford.edu/people/widom/paper-writing.html`. Accessed: 2025-04-20.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. 2025. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *Preprint*, arXiv:2504.08066.

Jinghui Zhang, Dandan Qiao, Mochen Yang, and Qiang Wei. 2024. Regurgitative training: The value of real data in training large language models. *arXiv preprint arXiv:2407.12835*.

Zining Zhang, Bingsheng He, and Zhenjie Zhang. 2023. Harl: Hierarchical adaptive reinforcement learning based auto scheduler for neural networks. In *Proceedings of the 51st International Conference on Parallel Processing*, ICPP '22, New York, NY, USA. Association for Computing Machinery.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Zekun Zhou, Xiaocheng Feng, Lei Huang, Xiachong Feng, Ziyun Song, Ruihan Chen, Liang Zhao, Weitao Ma, Yuxuan Gu, Baoxin Wang, and 1 others. 2025. From hypothesis to publication: A comprehensive survey of ai-driven research support systems. *arXiv preprint arXiv:2503.01424*.

Minjun Zhu, Yixuan Weng, Linyi Yang, and Yue Zhang. 2025. Deepreview: Improving llm-based paper review with human-like deep thinking process. *Preprint*, arXiv:2503.08569.

A Related Work

LLMs Assist Academic Writing Research on LLMs for academic writing falls into four primary categories. First, *automated paper generation* attempts to produce complete papers but often lacks user control and academic rigor (Shao et al., 2024; Jiang et al., 2024; Weng et al., 2025; Asai et al., 2024; Schmidgall et al., 2025). Second, *research ideation* employs LLMs to propose novel ideas and methodologies, though concerns regarding authorship and originality persist (Baek et al., 2024; Ghafarollahi and Buehler, 2024; Li et al., 2024a; Si et al., 2024). Third, thanks to the success of retrieval by instruction (Sun et al., 2024), *automated reviewing and research question answering* assist in literature searches and manuscript evaluations but do not directly refine writing quality (D'Arcy et al., 2024; Liang et al., 2024; Lu et al., 2024; Tan et al., 2025; Asai et al., 2024; Chen et al., 2024b; Lála et al., 2023; Song et al., 2024; Lin et al., 2024). Lastly, *LLM-assisted writing tools* enhance grammar and style and (Shi et al., 2023) improves a small paragraph of paper, they lack deep contextual awareness necessary for high-quality academic discourse (CoWriter, 2025; van Zeeland, 2023).

LLMs Assist Research Beyond writing, LLMs are increasingly utilized in autonomous research. (Swanson et al., 2024) introduced LLM agents functioning as research assistants, integrating human feedback into scientific workflows. ChemCrow (M. Bran et al., 2024) and Coscientist (Boiko et al., 2023) highlight LLM-led ideation and experimentation in chemistry, while ResearchAgent (Baek et al., 2024) automates research generation, iterative refinement, and review. AI Scientist (Lu et al., 2024) extends automation to coding, experimentation, and manuscript review. Despite these advancements, studies caution that LLMs require human oversight to ensure reproducibility and scientific rigor (Si et al., 2024).

Gaps and Contributions LLMs also contribute to research tasks such as code generation (Chen et al., 2021; Nijkamp et al., 2022), literature search (Ajith et al., 2024; Kang and Xiong, 2024; Press et al., 2024; Li et al., 2024b), and automated paper reviewing (D'Arcy et al., 2024; Liang et al., 2024; Lu et al., 2024; Weng et al., 2025). While they support ideation (Si et al., 2024), concerns about reduced creativity and homogenization persist (Chakrabarty et al., 2024; Anderson et al., 2024). Hybrid human-LLM approaches are seen as the most effective way to enhance research workflows (Ashkinaze et al., 2024; Liu et al., 2024b; Padmakumar and He, 2024).

Recently, the controllable generation of LLMs have been emphasized (Ge et al., 2025). While much work has focused on using LLMs for idea generation, review, and automation, little research directly addresses refining research papers to enhance coherence, clarity, and adherence to academic standards. Our work bridges this gap by leveraging LLMs specifically for structured refinement, allowing researchers to focus on deeper reasoning tasks while ensuring scholarly rigor.

LLM simulation Researchers have increasingly utilized Large LLMs to construct simulations, treating LLM agents as proxies for humans to perform actions and interactions (Park et al., 2023; Lin et al., 2023; Kong et al., 2024; Wang et al., 2024). These simulations have shown promise in diverse fields such as society, economics, policy, and psychology (Park et al., 2023; Li et al., 2024c; Chen et al., 2024a), while also serving as data generators and evaluators for LLM training (Tang et al., 2024; Zhang et al., 2024). However, LLMs face significant limitations in simulation tasks. Studies (Ai et al., 2024; Petrov et al., 2024; Hu and Collier, 2024; Lee et al., 2024) highlight their inability to maintain contextual consistency and produce fine-grained outputs. For example, Lee et al. (2024) found that LLMs exhibit consistent values and preferences even when role-playing diverse personas, underscoring their lack of adaptability and nuanced understanding.

B Prompts

Figure 4 shows the prompt for QA.

C ICLR 2024 Token Distribution

ICLR 2024 token distribution (derived from markdown) is showed in Figure 8.

The Prompt for QA

Act as an expert model for improving articles **PAPER_CONTENT**.
<SELECTED_CONTENT>
User Selected
</SELECTED_CONTENT>
<QUESTION>
<User Question>
</QUESTION>

Figure 4: Prompt for QA

D Data Quality Validation Details

Throughout the data collection and generation process, we applied strict quality control procedures to ensure the reliability and utility of the dataset for both training and evaluation. In particular, we conducted a human evaluation to verify that the revisions are applicable and effective across papers with diverse writing styles and technical focuses spanning multiple research domains.

Three Ph.D. students in computer science (referred to as Judges 1–3) were recruited as evaluators. Each annotator was assigned between 60 and 100 instruction–revision pairs randomly sampled from `ReviseQA`. Each revision is labeled according to the criteria defined in Figure 15, focusing on 4 key aspects: **(1) Instruction Following:** Whether the revision correctly follows the given instruction. **(2) Criteria Alignment:** Whether the revised text improves the original content based on the predefined section-level criteria (Table 2). **(3) In-Context Reference:** Whether the output appropriately references relevant information within the selected paragraph or surrounding context. **(4) Revision Acceptance:** Whether the revision is compelling enough for the evaluator to prefer it over the original.

Evaluation results in Table 15 show that the revisions are consistently effective across domains, supporting the validity of `ReviseQA` as a high-quality resource for instruction-based writing revision.

E Human Label Details

Table 8,9 shows the details of calculation data on human evaluation.

QA Controllability Assurance	**Judge 1**	**Judge 2**	**Judge 3**
GPT-4o-Mini -Instruction Following	(78+77+72+78+78)/5	(76+68+79)/3	(75+81+80+78+73)/5
-Criteria Following	(79+74+63+77+75)/5	(77+68+79)/3	(76+81+77+74+75)/5
-In-Context Ability	(73+53+48+62+61)/5	(67+57+75)/3	(69+76+74+73+72)/5
-Agree revision?	(48+48+44+53+53)/5	(65+56+64)/3	(67+74+74+72+71)/5
—-			
GPT-o1-mini -Instruction Following	(79+71+76+76+80)/5	(79+81+77)/3	(75+80+78+77+64)/5
-Criteria Following	(72+70+74+74+80)/5	(79+77+75)/3	(74+80+78+76+63)/5
-In-Context Ability	(74+53+58+65+60)/5	(68+68+68)/3	(73+75+81+75+64)/5
-Agree revision?	(58+50+53+59+60)/5	(66+66+67)/3	(72+76+78+76+62)/5

Table 8: Human evaluation on improvement acceptance rates before and after paragraph. we ask 3 human evaluators based on 5,3,5 paper, about 100,60,100 questions in score 1-5. The **Aggregated** column aggregates the results of 3 human evaluators.

F Section-Level Criteria Details

Section-level criterias are detailed in Figure 9,10,11,12,13,14.

PaperCursor	Judge 1	Judge 2	Judge 3
Instruction Following	(62+61+72+76+74)/5	(80+80+79)/3	(86+80+83+82+78)/5
Criteria Following	(60+60+69+72+73)/5	(74+74)/3	(82+82+82+81+82)/5
In-Context Ability	(58+51+48+61+61)5	(67+69)/3	(85+80+82+79+80)/5
Agree revision?	(50+45+44+55+52)/5	(65+64)/3	(83+79+82+80+77)/5

Table 9: PaperCursor Human Evaluation

Table 11: Average output length of each model on the 7,000-instance ReviseQA test set. Some models tend to produce significantly longer responses, occasionally including unsolicited explanations.

Models	Title	Abstract	Introduction	Background	Evaluation	Conclusion
Original Text	14.5	174.9	200.1	238.9	170.8	151.3
Phi-4	126.8	265.2	339.5	357.1	348.3	329.6
DeepSeek-v3-671B	17.5	158.7	243.5	258.8	245.7	217.4
GPT-4o-Mini	16.9	189.4	254.4	265.8	227.9	210.9
Qwen-2.5-7B-Instruct	21.2	187.3	272.2	258.4	264.1	242.9
Qwen-QWQ-32B	40.7	224.1	272.9	265.4	308.7	245.5
Llama-3.2-3B-Instruct	327.4	445.2	464.4	479.3	472.3	443.7
Qwen2.5-1.5B-Instruct	89.5	291.8	316.2	336.8	361.4	352.9
Qwen2.5-72B-Instruct	16.2	195.1	275.6	284.2	305.9	246.3
PaperCursor-14B	15.9	173.8	231.0	251.6	242.3	205.5
PaperCursor-7B	15.5	180.2	233.1	251.2	250.1	213.5
PaperCursor-3B	15.9	179.7	237.8	254.2	251.8	214.8
PaperCursor-1.5B	15.7	183.0	232.4	248.9	251.6	213.1
GPT-3.5-Turbo	57.9	144.0	178.3	189.8	169.2	146.4
DeepSeek-R1-Distill-Qwen-7B	419.1	668.5	890.3	942.3	1081.2	780.7

G Addressing Length Bias in LLM Judges.

While win rate is a commonly used metric for evaluating model performance in paragraph rewriting tasks, it becomes unreliable in the presence of length bias, a phenomenon well-documented in prior work, where LLM judges tend to be length-biased, preferring longer responses over shorter ones (Koo et al., 2024; Shen et al., 2023). Given the typical variation in output lengths across models (as shown in Table 11), the **length-controlled win rate** (Dubois et al., 2024) should be considered to ensure fairer comparisons and avoid gaming the metric with belabored points. This metric explicitly adjusts for length differences, mitigating the bias introduced by verbosity. Our approach is further supported by recent evaluation protocols such as AlpacaEval (Tatsu-lab, 2023), and enables a more accurate assessment of the true quality of model-generated revisions.

H Hyperparams

Hyperparameter	value
Batch Size	{1,2}
Cut-off Len	16384
max_new_tokens	512
Epoch	{10,20}
Learning Rate	{1e-5,2e-5}

Table 10: Hyperparameters

I Win Rate

Table 12 shows the win rate without length control, which is unreasonable compared to Table 3.

Models	Title	Abstract	Introduction	Background	Evaluation	Conclusion	Average↑
Qwen2-72B-Instruct	53.57	70.93	77.52	86.76	91.90	73.71	75.73
GPT-4o-Mini	65.57	59.71	70.05	67.81	70.86	62.14	66.02
Qwen-QWQ-32B-Preview	62.97	66.42	61.33	73.24	72.48	74.29	69.88
Deepseek-V3-671B (Liu et al., 2024a)	63.79	59.29	66.19	61.24	88.95	58.57	66.33
Qwen-2.5-7B-Instruct	60.79	70.93	60.52	56.48	74.48	70.43	65.60
PaperCursor (base: `Qwen-2.5-7b-instruct`) (*anchor*↑)							
Llama-3.1-8B-Instruct	47.41	39.64	41.24	55.24	55.71	30.29	44.92
Qwen2.5-1.5B-Instruct	34.36	32.39	26.14	21.24	26.48	31.29	28.65
GPT-3.5-Turbo	28.57	20.79	19.38	20.95	23.05	11.43	20.70
Llama-3.2-3B-Instruct	27.43	9.29	10.90	9.71	14.67	6.43	13.07

Table 12: Win rates of various models against `PaperCursor` (*anchor*) across different evaluation categories. Models are ranked in descending order based on their averaged win rates.

J Annotators for Controllable Quality Assurance

Figure 15 show annotators for controllable quality assurance.

K Case Study

We chose HARL (Zhang et al., 2023) in Figure 17 as a case study to demonstrate the application of `PaperCursor` in human-AI collaboration. `PaperCursor` helps the author refine the paper in a controllable manner.

L Complete Case in Table 4

Table 13 shows the complete case in Table 4 comparing GPT-4o and `PaperCursor`.

Shared Content (from abstract): ... To achieve better generalizability of learned graph representation, we innovatively propose a **C**onstrained **V**ariational **G**eneration (CVG) framework to generate generalizable graphs. Our framework is implemented based on the Variation Graph Auto-Encoder (VGAE) structure and optimized under the guidance of the Graph Information Bottleneck (GIB) principle, with its effectiveness validated by our theoretical analysis. We conduct extensive experiments on real-world datasets and demonstrate the superiority of our framework over state-of-the-art baselines. **Question:** How can we effectively summarize our key innovations and their contributions to the field of Generalizable Graph Learning? What specific results and comparisons to existing methods should we include to demonstrate the significance of our findings? how the experiments reflect the effectiveness of our framework, for example improve how many percent on which task?	
GPT-4o Response:	**`PaperCursor` Response:**
... To tackle this, we propose a method that not only identifies candidate reward functions but also ensures that the selected reward functions are aligned with the underlying task. We establish the conditions under which PAGAR can guarantee successful task completion, thereby providing a robust framework for enhancing the reliability of IRL-based IL systems. Our experimental results demonstrate that PAGAR significantly outperforms existing baselines across various complex and zero-shot IL tasks, showcasing its effectiveness in mitigating reward misalignment.	... To achieve better generalizability of learned graph representation, we innovatively propose a **C**onstrained **V**ariational **G**eneration (CVG) framework to generate generalizable graphs. Our framework is implemented based on the Variation Graph Auto-Encoder (VGAE) structure and optimized under the guidance of the Graph Information Bottleneck (GIB) principle. The key innovations of our CVG framework include: Constrained Variational Generation: By leveraging the VGAE structure, our framework generates graphs that are constrained to maintain essential invariant properties while allowing flexibility to adapt to unseen domains. Graph Information Bottleneck (GIB) Optimization: The GIB principle guides the optimization process to ensure that the generated graphs retain the most informative features for classification while discarding irrelevant variations. Our theoretical analysis validates the effectiveness of these innovations, ensuring that the learned representations are robust to distribution shifts.
Strengths: *Holistic Processing, e.g. refer to the exp.*; *Critical feedback*; *Without spoiling the flow of thought*; *Replace back to paper automatically*.	**Weaknesses:** *No length control*;*Simply paraphrase*; *Generalised description*; *Single interaction, lack of context*; *Like chatbot, not human-ai collaboration.*

Table 13: Case study comparing GPT-4o and `PaperCursor`.

M Baseline Model Details

Table 14 shows the baseline model details.

N Controllability Annotation Criterias and Interface

To ensure our data and model quality, We invited three AI experts specializing in inference speedup, graph neural networks (GNN), and Field Programmable Gate Arrays (FPGA) to annotate 5, 3, and 5 papers, respectively. Each paper includes 20 question-answer pairs per model, focusing on section-level improvements. These pairs are distributed across different sections of the paper as follows: 2 for the

Models	Backbone	GitHub Stars	Huggingface Download
Closed-Source			
GPT-4-Turbo (OpenAI, 2024)		/	
GPT-4o-mini (OpenAI, 2024)		/	
GPT-3.5-Turbo (Brown et al., 2020)		/	
Open-Source (> 8B)			
Deepseek-R1 (DeepSeek-AI, 2025)	Llama	50K	285K
Deepseek-V3-671B (Liu et al., 2024a)	DeepSeek-V3-Base	63K	374K
Deepseek-V3-32B (Liu et al., 2024a)	DeepSeek-V3-Base	63K	374K
Qwen-2-72B-Instruct (et.al, 2024b)	Qwen-2-72B-Instruct	45.3K	374K
QwQ-32B-Preview (Team, 2024)	Qwen2.5-32B-Instruct	15K	198K
Phi-4 (14B) (Abdin et al., 2024)	-	-	557K
Open-Source (≤8B)			
Llama-3.1-8B-Instruct (et.al, 2024a)	Llama-3.1-8B-Instruct	28.1K	5.75M
Qwen-2.5-7B-Instruct (et.al, 2024b)	Qwen-2.5-7B	12.6K	1.27M
Llama-3.2-3B-Instruct (et.al, 2024a)	Llama-3.2-3B	28.1K	1.48M
Qwen-2.5-1.5B-Instruct (et.al, 2024b)	Qwen-2.5-1.5B	12.6K	551K

Table 14: Details information of baseline models. Data collected at 30.1.2025. The "/" indicates that the model uses a private download link or that its download statistics on HuggingFace are not disclosed.

title, 4 for the abstract, 6 for the introduction, 3 for the background, 3 for the evaluation, and 2 for the conclusion. The controllable criteria used for evaluation are presented in Figure 15. The annotators' operating interface and the interface of `PaperCursor` are listed in Figure 16.

O Can LLMs Score Full Papers?

In the context of academic paper evaluation, the only available human expert review labels at full-paper granularity come from OpenReview. Unfortunately, due to the high cost and inherent biases of human reviews—evidenced by a standard deviation of 1.26 in reviewer ratings for each paper in 2024— it is impractical to invite expert reviewers for every benchmarking scenario that requires full-paper scoring.

To address this limitation, several studies (Lu et al., 2024; Weng et al., 2025; Tan et al., 2025) have explored the use of LLMs for predicting full-paper scores. A key question remains: are LLMs suited to be a reliable reviewer? To investigate this, we follow the approach of (Lu et al., 2024), applying NeurIPS review guidelines and few-shot examples to assess our test set.

As shown in Figure 18, scaling up model parameters is significantly more challenging for paper scoring compared to MMLU-Pro. We can infer that the bottleneck in the paper scoring task **cannot be simply solved by scaling the model**. LLMs struggle with paper scoring, which is already quite challenging even for human experts (1.16 rating MAE per paper according to (Weng et al., 2025)).

Rather, it emphasizes long-context understanding and localized revision, areas in which GPT-4o-mini performs reliably. As validated by human annotators (see Table 15), GPT-4o-mini produces coherent and high-quality paragraph-level revisions suitable for this task. The full set of criteria is illustrated in Figures 9 through 14, and the prompt design is detailed in Appendix 5.

Table 15: Human evaluation of improvement acceptance rates before and after revision. Three human evaluators assessed 5, 3, and 5 papers respectively, scoring 100, 60, and 100 instruction–revision pairs using a 1–5 scale. The **Aggregated** column reports the average scores across all three evaluators.

Criterion	Judge 1	Judge 2	Judge 3	Aggregated
GPT-4o-Mini				
-Instruction Following	3.83	3.72	3.87	3.81
-Criteria Alignment	3.68	3.74	3.83	3.75
-In-Context Reference	2.97	3.32	3.64	3.31
-Revision Acceptance	2.46	3.09	3.58	3.04
GPT-o1-mini				
-Instruction Following	3.82	3.95	3.74	3.84
-Criteria Alignment	3.70	3.85	3.71	3.76
-In-Context Reference	3.10	3.40	3.68	3.40
-Revision Acceptance	2.80	3.32	3.64	3.25

The Prompt for Generating QA pairs

You are an advanced language model designed to assist users in improving their articles. Users will provide an article in LaTeX or Markdown format and specify a **section** along with **criteria** for improvement. Your task is to identify a specific selected content from the provided section, align it with the given criteria, and offer actionable feedback to improve the content.
Instructions:
1. **Role 1**: We have a paper improvement task with a specific criteria 'criteria['prompt']'. Now play a role as an author of the provided paper content. Select a specific content from the section 'section' (or equivalent), and ask a chatbot assistant to help you improve that selected content.
- **The selected paper content must be a worth-improving paragraph(s)** that might not achieve the standards of the criteria 'criteria['prompt']', and that content should come from the section 'section'. The selected content will be labeled as **BEFORE IMPROVEMENT**.
- Provide a concise, conversational improvement-related question labeled as **QUESTIONS**. These questions should not explicitly tell what rules or standards to follow or what the specific goal should be. Instead, offer a high-level instruction that may hint at the criteria without stating them directly. The aim is to allow for creativity and subtle alignment with the criteria.
- Keep the question short and conversational.
2. **Role 2**: Act as an expert model for improving articles.
The revised version of the selected content should be labeled as AFTER IMPROVEMENT and specifically address the QUESTIONS on BEFORE IMPROVEMENT above. Avoid adding unnecessary length, unrelated details, overclaims, or vague statements. Focus on clear, concise, and evidence-based improvements that align with the overall context of the paper.
Provide a detailed explanation of the changes made, labeled as EXPLANATION, with clear references to the paper's content. Ensure the explanation demonstrates how the revisions align with the context and criteria of the paper.
— PAPER CONTEXT STARTS
paper_latex
— PAPER CONTEXT ENDS
Response Format (must be strictly followed):
— BEFORE IMPROVEMENT STARTS
<Selected content>
— BEFORE IMPROVEMENT ENDS
— QUESTIONS START
<Concise, improvement-related question based on the criteria 'criteria['prompt']'>
— QUESTIONS END
— AFTER IMPROVEMENT STARTS
<Revised version of the selected content to answer the **Questions** above> — AFTER IMPROVEMENT ENDS
— EXPLANATION STARTS
<An explanation of the changes made, showing how they align with the context of the article and address the criteria. Include references from the paper context where relevant.>
— EXPLANATION ENDS

Figure 5: Prompts for Generating `ReviseQA`

The Prompt for Judging

You are a highly efficient assistant, who evaluates and rank large language models (LLMs) based on the quality of their responses to given prompts. This process will create a leaderboard reflecting the most accurate and human-preferred answers.
I require a leaderboard for various large language models. I'll provide you with prompts given to these models and their corresponding responses. Your task is to assess these responses, ranking the models in order of preference from a human perspective. Once ranked, please output the results in a structured JSON format for the make_partial_leaderboard function.
Prompt

```
{
    "instruction": "{instruction}",
}
```

Model Outputs
Here are the unordered outputs from the models. Each output is associated with a specific model, identified by a unique model identifier.

```
{
    {
        "model": "m",
        "output": "{output\_1}"
    },
    {
        "model": "M",
        "output": "{output\_2}"
    }
}
```

Task
Evaluate based on the quality and relevance to the instructions. The following is the definition of the quality on the section `<section>`: `<criteria["prompt"]>`. If the model's output refers to information beyond `<Selected content>`, it receives a slightly higher score.

Figure 6: Prompts for Judging (modified from alpaca_eval_gpt4_turbo_fn).

The Prompt for ranking

Human: I want you to create a leaderboard of different large-language models. To do so, I will give you the instructions (prompts) given to the models, and the responses of two models. Please rank the models based on which responses would be preferred by humans. All inputs and outputs should be Python dictionaries.
Here is the prompt:

```
{
    "instruction": "{instruction}",
}
```

Here are the outputs of the models:

```
{
    "model": "model_1",
    "answer": "{output_1}"
},
{
    "model": "model_2",
    "answer": "{output_2}"
}
```

Now please rank the models by the quality of their answers, so that the model with rank 1 has the best output. Then return a list of the model names and ranks, i.e., produce the following output:

```
[
    {'model': \texttt{<model-name>},
    'rank': \texttt{<model-rank>}},
    {'model': \texttt{<model-name>},
    'rank': \texttt{<model-rank>}}
]
```

Your response must be a valid Python dictionary and should contain nothing else because we will directly execute it in Python. Please provide the ranking that the majority of humans would give.

Figure 7: Prompts for Scoring.

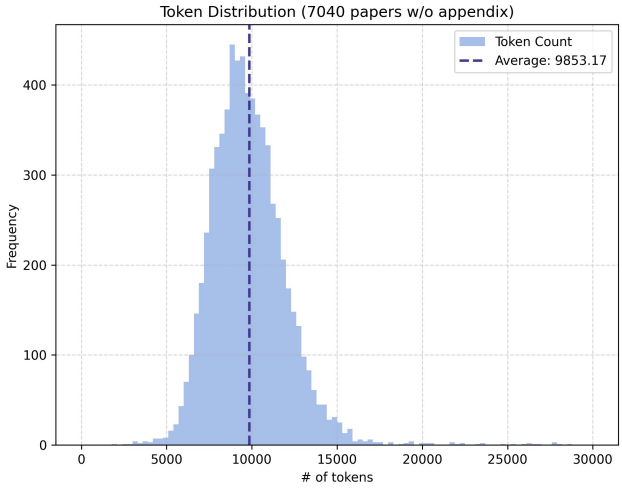


Figure 8: ICLR 2024 Paper Token Distribution (without Appendix)

Criteria Details of Section Title

1. Consistency and Alignment of Title with Paper's Content:
Evaluate the degree to which the paper's title accurately captures its principal topics, arguments, or findings. Does the title reflect the scope and focus of the paper, and is it consistent with the main concepts and keywords presented in the abstract and introduction? Identify any discrepancies or misalignment between the title and the content.

2. Conciseness and Clarity of Title:
Evaluate the paper's title for redundancy. Are there repeated words or concepts that could be removed without changing the core meaning? Does the final title remain succinct, clear, and accurately convey the paper's main focus or contribution?

Figure 9: Criteria Details of Section Title

Criteria Details of Section Abstract

1. Clarity and Impact of the Conclusion:
Evaluate the clarity and impact of the conclusion in the abstract. Does it clearly summarize the research steps, highlight key outcomes, and explain the significance of these outcomes for the field of computer science? Are the primary technical advancements and their contributions presented in a concise and unambiguous manner?

2. Motivation and Purpose in the Abstract:
Evaluate how well the abstract communicates the research's motivation. Does it clearly articulate the broader issue, concept, or problem in Computer Science that the work addresses? Does it explicitly state the specific research problem being solved and why it is important?

3. Explanation of Existing Solutions and Research Gap:
Assess how well the abstract explains the shortcomings of current solutions and highlights the corresponding research gap. Does it clearly articulate why existing methods are insufficient and how the proposed approach addresses these limitations? Is the explanation comprehensible to a wide audience, from domain experts to non-specialists?

4. Clarity and Adequacy of Proposed Solutions:
Assess how effectively the abstract communicates the proposed solutions. Does it clearly identify the research gap or problem being addressed, and explain how the proposed solution tackles this gap? Does it highlight the novelty or contribution of the solution, demonstrating its relevance or improvement over existing work? Rate the clarity, completeness, and significance of the explanation provided in the abstract.

Figure 10: Criteria Details of Section Abstract

Criteria Details of Section Introduction

1. Strength and Clarity of Motivation in the Introduction:
Evaluate whether the motivation in the Introduction is specific and convincing. Does the paper avoid over-generalization and clearly articulate the significance of the issue? Are concrete examples, statistics, or contextual details used to establish why the problem matters?

2. Review of Existing Approaches in Introduction:
Assess the thoroughness and clarity of the literature review within the introduction. Does the paper cite and critique relevant prior works, highlighting both their methodologies and limitations? Does the introduction establish how the proposed work builds upon or differentiates itself from existing research, and is there sufficient context provided to demonstrate the significance of the current study? Are any quantitative or qualitative comparisons made where appropriate?

3. Audience Alignment and Appropriateness:
Evaluate whether the introduction is effectively tailored to its target audience. Is the complexity, depth, and choice of terminology suitable for the presumed background knowledge of the readership? Does the introduction provide sufficient context without oversimplifying or overwhelming the intended audience?

4. Clarity and Visibility of Contributions:
Assess the clarity and visibility of the paper's contributions. Are the core contributions explicitly stated in a dedicated paragraph or section toward the end of the introduction? Are they understandable to a broad scientific audience, presented succinctly, and positioned logically following the problem statement and background information?

5. Clarity and Specificity of Problem Definition:
Evaluate the paper's problem definition in terms of four key elements: current situation, ideal situation, the gap between them, and how the research aims to address this gap. Are these components clearly stated, distinct, and directly tied to the research objectives? Does the definition provide sufficient clarity and focus for the research?

6. Integration of State-of-the-Art in Problem Framing:
Evaluate how effectively the introduction incorporates the State-of-the-Art (SOTA) to frame the research problem. Does it include explicit references to key works, methodologies, or findings that highlight relevant gaps or limitations in the field? Is there a clear logical link between the SOTA discussion and the stated research objectives, demonstrating how the proposed work builds upon or extends existing research?

Figure 11: Criteria Details of Section Introduction

Criteria Details of Section Background

1. Contextual Relevance and Clarity of Background:
Assess how effectively the background section establishes context for the research. Does it provide a clear overview of the broader field in computer science, then narrow down to the specific problem? Does the paper avoid making unwarranted assumptions about the reader's prior knowledge? Finally, does it clarify why addressing the problem is important to the field?

2. Coverage of Key Preliminary Concepts:
Evaluate the thoroughness and clarity of the paper's background or preliminary section. Does it introduce and define all the critical concepts, algorithms, or theorems necessary to understand the technical contributions? Are these concepts clearly explained, logically organized, and accessible to readers who are not experts in the field? Does the paper use consistent terminology and adequately explain symbols, abbreviations, or specialized terms before their first usage?

3. Clarity and Consistency of Terminology:
Assess the clarity and consistency of the key terms introduced in the background section. Are all critical terminologies defined at their first occurrence and used consistently throughout the paper? Does the paper avoid undefined shifts or redefinitions of terms, and does it align terminology with standard conventions in the field?

Figure 12: Criteria Details of Section Background

Criteria Details of Section Evaluation

1. Experimental Setup Clarity and Reproducibility:
Evaluate how clearly and thoroughly the experimental setup is described. Does the paper provide all necessary information on hardware/software configurations, parameter settings, data preprocessing steps, and any contingency procedures, such that others could replicate the experiments with the same resources?

2. Depth and Clarity of Figures and Tables Analysis:
Evaluate the thoroughness and clarity of the paper's analysis of figures and tables. Are the data clearly explained and linked to the research objectives or hypotheses? Do the authors discuss trends, patterns, or anomalies, and interpret quantitative metrics in a way that highlights their significance? Is there a clear comparison to baselines or related work, demonstrating how the results fit into or advance the field? Do the authors emphasize key takeaways and practical or theoretical implications arising from the findings?

3. Experimental Support for Main Innovations:
Evaluate how thoroughly the paper's main innovations or contributions are backed by experimental evidence. Does the paper provide direct tests or comparisons to validate each innovation? Are quantitative or qualitative results clearly linked to the claims made, with appropriate metrics and comparisons against baselines or existing methods? Are ablation studies or sensitivity analyses included to demonstrate the significance of each component? If certain claims are not experimentally supported, have the authors either provided additional experiments or adjusted their claims accordingly?

Figure 13: Criteria Details of Section Evaluation

Criteria Details of Section Conclusion

1. Broader Impact and Future Directions:
Assess the thoroughness of the paper's conclusion or discussion sections in addressing the broader impact of the research. Does the paper provide specific and clear avenues for future work?

2. Clarity and Impact of Key Innovations and Findings:
Evaluate whether the conclusion effectively highlights the paper's key innovations.

Figure 14: Criteria Details of Section Conclusion

Criteria

Each QA pair is evaluated based on four metrics, each scored from 1 to 5:
Evaluation Metrics (1-5 Scoring Criteria)
1. **Instruction Following:** Evaluate whether the answer correctly follows the given instruction.
1 – The answer completely ignores or contradicts the instruction.
2 – The answer only partially follows the instruction, with major missing elements.
3 – The answer follows the instruction but lacks completeness or clarity.
4 – The answer mostly follows the instruction with minor inconsistencies.
5 – The answer strictly follows and fully satisfies the instruction.
2. **Criteria Following:** Evaluate whether the revised text improves the original content based on predefined criteria.
1 – The revision does not follow any criteria and worsens the content.
2 – The revision attempts to follow the criteria but makes the content unclear.
3 – The revision follows the criteria but does not provide a significant improvement.
4 – The revision improves clarity and correctness while adhering to the criteria.
5 – The revision strictly follows the criteria and significantly improves the original content.
3. **In-Context Ability:** Evaluate whether the model's output appropriately references information within Selected Content.
1 – The output ignores Selected Content and adds irrelevant external information.
2 – The output relies on external information without justification.
3 – The output primarily references Selected Content but includes minor unrelated details.
4 – The output correctly refers to Selected Content with minimal external additions.
5 – The output strictly remains within Selected Content while providing a relevant and precise response.
4. **Agree Revision:** Evaluate whether the revision is convincing enough for the user to adopt it as a replacement.
1 – The revision is clearly worse than the original text.
2 – The revision is slightly better but has major flaws, making it unlikely to be adopted.
3 – The revision is neutral or slightly better, but adoption is uncertain.
4 – The revision is clearly better, and most users would likely adopt it.
5 – The revision is significantly better, and users would confidently adopt it.

Figure 15: The criteria for human instructors.

```
1 This work proposes HARL, a hierarchical and adaptive
  reinforcement learning-based auto-scheduler for tensor
  programs. It is templatefree with improved performance
  on the generated codes and significantly faster tuning
  process. The effectiveness of HARL is verified
2 on different kinds of tensor operators and DNNs, under
  CPU and GPU hardware platforms. We believe that, HARL
  is an innovation on how to utilize the advanced
  reinforcement learning based
3 decision process to facilitate the time-consuming
  tensor code optimization.
```

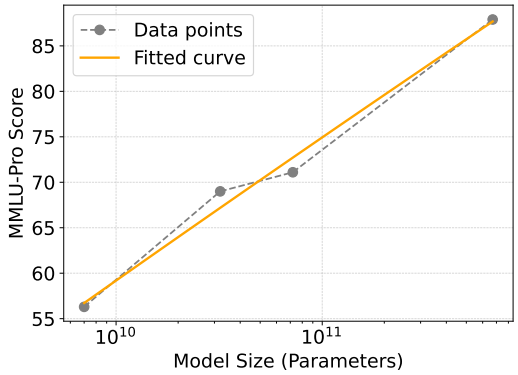
```
1 In this paper, we introduce HARL, a hierarchical and
  adaptive reinforcement learning-based auto-scheduler
  that significantly enhances tensor program exploration.
  By employing a hierarchical RL architecture, HARL
  enables learning-based decisions at various levels of
  search granularity, leading to a $22\%$ improvement in
  tensor operator performance and a $4.3\mathrm{x}$
  increase in search speed compared to the state-of-the-
  art auto-scheduler, Ansor. Our experiments reveal that
  HARL not only outperforms Ansor in both tuning speed
  and final program performance but also achieves up to
  $8\%$ improvement in inference performance on end-to-
  end neural networks. These results underscore the
  effectiveness of HARL in addressing the challenges of
  neural network optimization, providing a robust
  framework for future research in this domain.
```

Figure 16: A comparison of the paragraph before and after revision.

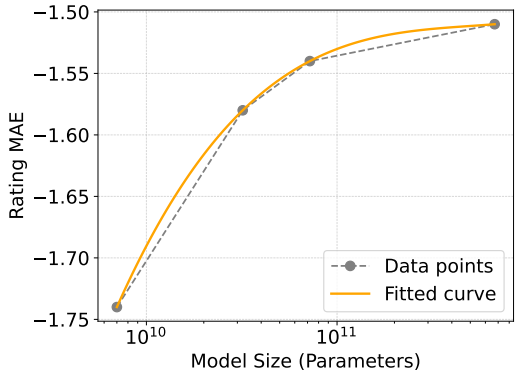
Selected Content: This work proposes HARL, a hierarchical and adaptive reinforcement learning-based auto-scheduler for tensor programs. It is templatefree with improved performance on the generated codes and significantly faster tuning process. The effectiveness of HARL is verified on different kinds of tensor operators and DNNs, under CPU and GPU hardware platforms. We believe that, HARL is an innovation on how to utilize the advanced reinforcement learning based decision process to facilitate the time-consuming tensor code optimization.
User Interaction: How can we effectively summarize the key innovations of HARL and their contributions to the field of neural network optimization? What specific results and comparisons to existing methods should we include to demonstrate the significance of our findings?

Responses: In this paper, we introduce HARL, a hierarchical and adaptive reinforcement learning-based auto-scheduler that significantly enhances tensor program exploration. By employing a hierarchical RL architecture, HARL enables learning-based decisions at various levels of search granularity, leading to a 22% improvement in tensor operator performance and a $4.3\mathrm{x}$ increase in search speed compared to the state-of-the-art auto-scheduler, Ansor. Our experiments reveal that HARL not only outperforms Ansor in both tuning speed and final program performance but also achieves up to 8% improvement in inference performance on end-to-end neural networks. These results underscore the effectiveness of HARL in addressing the challenges of neural network optimization, providing a robust framework for future research in this domain.
Explanations: The revised conclusion emphasizes the key innovations of HARL, specifically highlighting the 22% performance improvement and $4.3\mathrm{x}$ speed increase over Ansor. By including quantitative results and comparisons to existing methods, we demonstrate the significance of our findings in the context of neural network optimization. The hierarchical RL architecture is clearly linked to these improvements, showcasing how it facilitates more efficient exploration of tensor programs. This alignment with the criteria of effectively summarizing key innovations and providing concrete results is crucial for advancing the field.

Figure 17: A use case on `PaperCursor`.



(a) MMLU-Pro scores on multi-task understanding across different # of model parameters



(b) MAE scores on paper scoring across different # of model parameters

Figure 18: Scaling trends of Qwen-2.5-7B/32B/72B/Max-Instruct performance. (a) MMLU-Pro scores stably improve with model size. Scaling is effective on multi-task understanding. (b) In the paper scoring task, the rating MAE struggles to go below 1.5. As the model size increases, the reduction in MAE becomes smaller, indicating that scaling offers limited performance improvement.

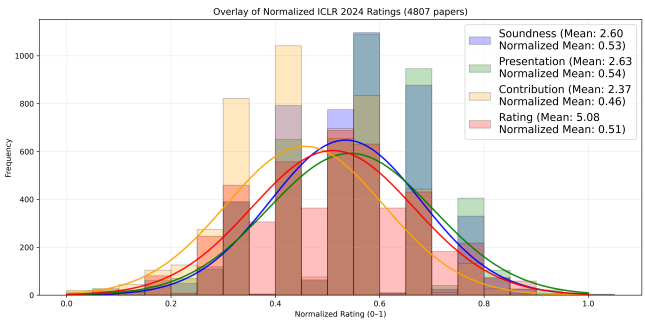


Figure 19: ICLR 2024 paper rating distribution.