RESEARCHARCADE: GRAPH INTERFACE FOR AUTOMATED RESEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

AutoResearch systems increasingly assist with literature discovery, scientific writing, peer review, and manuscript revision. However, they commonly rely on fragmented, task-specific views of academic data that fail to expose relations across research artifacts. In order to address this issue, we introduce RESEARCHARCADE, a graph interface that unifies multi-source, multi-modal, and temporally evolving academic data for diverse research models and workflows. RESEARCHARCADE organizes ArXiv papers and OpenReview peer-review activities in a coherent multi-table graph connecting text, figures, tables, citations, reviews, and paper revisions, and exposes task-relevant neighborhoods through a common interface. We evaluate RESEARCHARCADE on six academic tasks spanning the research process and two evidence-discovery workflows under both retrieval and agentic settings. Across six academic tasks, RESEARCHARCADE supports diverse predictive and generative models and enables tasks that combine cross-source and multi-modal information. In two additional AutoResearch workflows, controlled comparisons against degree-preserving rewired graphs show that correct citation connectivity consistently improves evidence discovery in both fixed retrieval pipelines and interactive agents. The results further demonstrate that graph-structured academic data enhances the evidence-discovery capabilities of AutoResearch systems. Together, these findings establish RESEARCHARCADE as a general graph interface for supporting diverse research models and workflows.

1 INTRODUCTION

Academic research represents a pinnacle of human knowledge discovery. As the scientific community grows, AutoResearch systems are increasingly used to support researchers throughout the research process (Lin et al., 2024; Liu et al., 2026). However, these systems are often built on separate, task-specific data pipelines that provide only partial views of the underlying research ecosystem and overlook relations among academic artifacts. This raises a central question: *Can a unified academic graph improve evidence discovery, and can controlled interventions identify where these gains come from?*

Addressing this question requires an interface that reflects how research knowledge is organized and evolves. Such an interface must integrate heterogeneous data sources, preserve relations across multi-modal and temporally evolving artifacts, and accommodate models with different input requirements. Meeting these requirements within a reusable representation remains an open challenge.

Developing a unified representation of research activities remains an open challenge. Existing academic datasets systematically organize scientific papers or peer reviews (Kang et al., 2018; Lo et al., 2019), but primarily focus on individual data sources. Some datasets incorporate multi-modal content (Xia et al., 2024; Tian et al., 2025), yet do not fully connect information across sources and artifact types. Recent work represents academic data and tasks as graphs (Li & Tajbakhsh, 2023; Zhang et al., 2024), but typically construct a separate graph and interface for each task. Consequently, they provide limited support for reusing the same academic structure across different models and AutoResearch workflows.

In this paper, we introduce RESEARCHARCADE, a graph interface that connects diverse academic data sources and supports both task-specific research models and interactive research agents. RESEARCHARCADE is designed around four properties essential to AutoResearch: *multi-source*, *multi-*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

modal, *structurally heterogeneous*, and *dynamically evolving*. It integrates papers from ArXiv with peer reviews and revisions from OpenReview, linking multi-modal academic content to paper's evolution process. These artifacts are organized in a coherent multi-table node–edge schema, which exposes their relations as task-relevant graph neighborhoods and retrieval operations. At the paper level, RESEARCHARCADE preserves revisions and peer-review histories; at the community level, its extensible design supports continuous data incorporation and time-aware analysis of research trends. A common task interface transforms these academic graphs into inputs for models with different requirements, supporting both predictive and generative tasks as well as interactive AutoResearch workflows.

To evaluate the breadth of RESEARCHARCADE, we study six academic tasks: citation prediction, paragraph generation, revision retrieval, revision generation, acceptance prediction, and rebuttal generation. These tasks span multiple stages of the research process and cover both predictive and generative models. We further evaluate two evidence-discovery workflows under retrieval and agentic settings. Across these workflows, correct citation connectivity improves evidence recall over both lexical retrieval and degree-preserving rewired controls. In fixed retrieval, the correct graph adds 29.1 and 17.5 recall points over the rewired graph on the two workflows. The advantage persists when graph access is placed inside agent loops, where correct connectivity adds 6.4–10.3 recall points across all evaluated models. At the same time, agents do not significantly outperform matched replays given the same observed candidate pools. Together, these results identify evidence access as the main source of the measured graph benefit.

Our main contributions are as follows. First, RESEARCHARCADE provides a unified and extensible data foundation for AutoResearch by connecting multi-source, multi-modal, and temporally evolving academic artifacts in a coherent multi-table graph. Second, RESEARCHARCADE exposes this data through a common graph interface that standardizes task formulation and provides task-relevant views for diverse predictive, generative, and agentic systems. Finally, controlled rewiring and matched-pool analyses use this interface to isolate where graph augmentation helps: correct connectivity expands the task-relevant evidence available to fixed pipelines and interactive agents, while matched candidate-pool replays perform comparably to the agents.

2 RELATED WORK

Graph-structured academic data. Existing work represents academic data as graphs at different levels of granularity. OAG-BENCH (Zhang et al., 2024) models academic communities using entities such as papers, authors, and affiliations. UNARXIVE (Saier et al., 2023) and DOCGENOME (Xia et al., 2024) provide finer-grained representations by decomposing papers into document-level components; the former emphasizes paragraph-level citations, while the latter incorporates multi-modal content. These resources primarily focus on individual corpora or task settings. In contrast, RESEARCHARCADE connects fine-grained paper content with community and peer-review activities in a reusable multi-table graph.

Dynamic modeling of academic data. Academic data evolve at two complementary scales: research communities change over time, while individual manuscripts undergo review and revision. At the community level, Gollapalli & Li (2015) analyze long-term topic evolution across ACL and EMNLP, while Tian et al. (2023) model splits and merges in scientific communities as event prediction. At the manuscript level, prior work aligns revisions at the sentence (Kuznetsov et al., 2022; D'Arcy et al., 2024) or paragraph level (Jourdan et al., 2025). Whereas these directions typically study the two scales separately, RESEARCHARCADE preserves both manuscript evolution and continuously updated community-level data within the same interface.

Models and agents for AutoResearch. Deep learning and language models are increasingly used for literature discovery, scientific writing, peer review, and other stages of the research process (Lin et al., 2024; Feng et al., 2024; Liu et al., 2026). Zhang et al. (2024) evaluate CNNs, GNNs, and LLMs across academic tasks. Recent agentic systems further combine language models with iterative search, tool use, and evidence synthesis to support multi-step literature review and end-to-end research workflows. However, these systems often rely on task-specific data construction and model-specific interfaces, limiting reuse across research stages. RESEARCHARCADE complements these systems by exposing fixed retrieval pipelines and interactive agents to the same temporally restricted academic graph, which enables controlled interventions on the structure available to both.

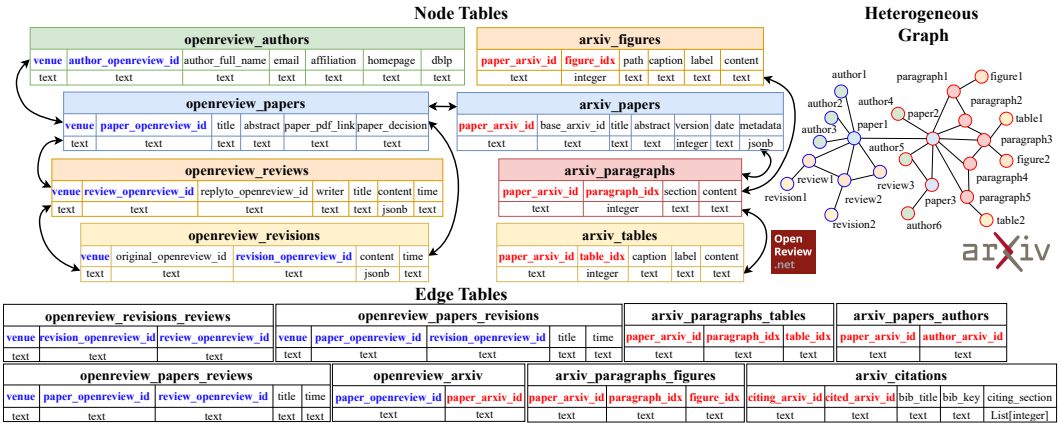


Figure 1: **Multi-table graph schema of RESEARCHARCADE.** Each table represents a node type (colored) or an edge type (black and white). The blue and red columns contain identifiers for entities and relations originating from OpenReview and ArXiv, respectively, while the remaining columns store their associated features. Author metadata contain email domains rather than email addresses.

3 RESEARCHARCADE DATA DESCRIPTION

RESEARCHARCADE provides a unified representation of real-world academic activities with four key properties: multi-source, multi-modal, highly relational and heterogeneous, and dynamically evolving. Figure 1 presents its multi-table graph schema, with additional details provided in Figure 3.

3.1 MULTI-SOURCE & MULTI-MODAL

RESEARCHARCADE draws primarily from research papers on ArXiv and peer-review records on OpenReview. In addition to textual content, it incorporates figures and tables, enabling models to access multi-modal scholarly information through the same interface. The complete data-construction procedure is described in Appendix A.2.

ArXiv. RESEARCHARCADE contains 66,918 papers spanning 11 scientific fields, including 569,501 sections, 8,014,095 paragraphs, 876,636 figures, and 324,648 tables, together with the relations among these entities. Detailed statistics and data-collection procedures are provided in Table 5 and Appendix A.2.1. RESEARCHARCADE also supports periodic updates through the continuous collection pipeline described in Appendix A.2.2.

OpenReview. RESEARCHARCADE contains 57,278 submissions to ICLR, NeurIPS, ICML, and EMNLP, involving 189,038 authors, 884,875 reviews, and 54,467 manuscript revisions collected during the review and rebuttal processes. It preserves the relations among submissions, authors, reviews, and revisions. Detailed statistics and collection procedures are provided in Tables 6–9 and Appendix A.2.3.

Connecting ArXiv and OpenReview. RESEARCHARCADE links OpenReview submissions to their corresponding ArXiv papers using title-based matching, connecting the public scholarly record with its review and revision history. In total, 25,969 OpenReview submissions (45.34%) are matched to ArXiv papers. Detailed statistics are reported in Tables 10–13.

3.2 HIGHLY RELATIONAL AND HETEROGENEOUS

Academic activities are represented through interactions among heterogeneous, typed entities. RESEARCHARCADE stores these entities and their relations in a multi-table node–edge schema that maps naturally to heterogeneous graphs, as illustrated in Figure 1.

For ArXiv, RESEARCHARCADE represents the scholarly literature at two complementary scales. At the intra-paper level, papers are decomposed into sections, paragraphs, figures, and tables connected by typed containment and reference relations. At the inter-paper level, papers are linked to authors, subject categories, and other papers through authorship, category-assignment, and citation relations. For OpenReview, RESEARCHARCADE represents the peer-review process by connecting submissions with authors, reviews, comments, and manuscript revisions.

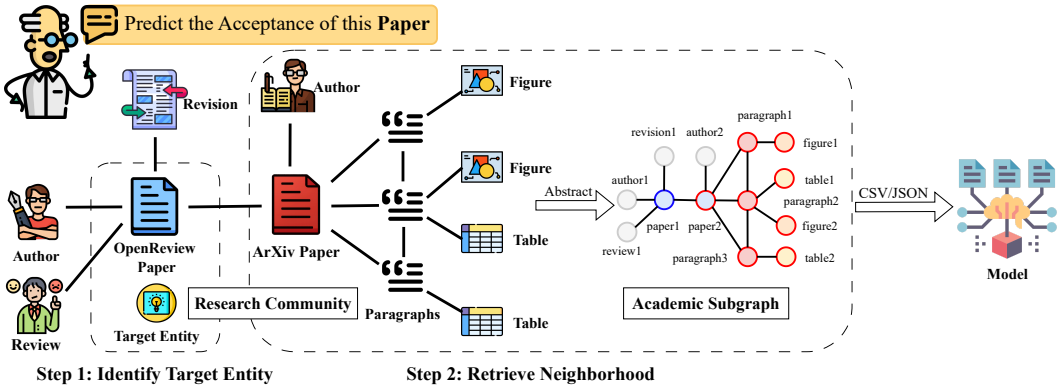


Figure 2: **RESEARCHARCADE provides a unified two-step scheme for defining academic tasks.** First, a target entity and one of its attributes are selected as the prediction target. Second, the target entity's graph neighborhood is retrieved to construct the task input.

Beyond providing a common storage format, this schema exposes task-relevant graph neighborhoods through a shared interface. As illustrated in Figure 2, task-specific models and interactive research agents can therefore access consistent views of the same underlying academic graph.

3.3 DYNAMICALLY EVOLVING

Academic knowledge and research activities evolve continuously. RESEARCHARCADE preserves temporal metadata such as paper submission dates, revision timestamps, and review events, supporting the study of scholarly evolution at both the paper and community levels. At the paper level, it captures how papers change through revision and rebuttal. At the community level, its extensible collection pipeline supports the incorporation of newly available academic records and the analysis of research trends over time.

By retaining timestamps and historical paper versions, RESEARCHARCADE also supports point-in-time graph construction, allowing models and agents to access only the academic information available at a specified time. This is particularly important for evaluating AutoResearch systems without introducing information from future papers, reviews, or revisions.

4 ACADEMIC TASKS AND AUTORESEARCH WORKFLOWS ON RESEARCHARCADE

Academic tasks require repeated data collection, preprocessing, and task-specific interface design. RESEARCHARCADE provides a common graph interface for two forms of research automation: task-specific models that operate on fixed graph views, and evidence-discovery systems that retrieve information through either fixed pipelines or interactive agents. Figure 2 illustrates the formulation of task-specific academic tasks. We then extend the same graph interface to support interactive evidence-discovery workflows.

4.1 ACADEMIC GRAPH AS A HETEROGENEOUS GRAPH

A heterogeneous graph can be defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$ and each edge $e \in \mathcal{E}$ is assigned a type through mapping functions. Specifically, the node type is defined by $\tau(v) : \mathcal{V} \to \mathcal{C}$, and the edge type is defined by $\phi(e) : \mathcal{E} \to \mathcal{D}$, where $c \in \mathcal{C}$ and $d \in \mathcal{D}$ represent the set of node types and the set of edge types. An edge e connecting a pair of nodes is denoted as $e = (v, u)$.

Data from RESEARCHARCADE can be represented as a heterogeneous academic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each node $v \in \mathcal{V}$ corresponds to a row in the node table, while each edge e corresponds to a row in the edge table. Furthermore, each node table V_c is associated with a unique node type c, and each edge table E_d is linked to a unique edge type d.

4.2 UNIFIED INTERFACE FOR ACADEMIC TASKS

As illustrated in Figure 2, RESEARCHARCADE defines task-specific academic tasks in two steps: identifying a target entity and constructing the graph view through which a model accesses relevant information. The same interface can expose temporally valid search and traversal operations to evidence-discovery systems.

Step 1: Specifying the research objective. For a task-specific academic task, the objective is defined by a target entity t, which may be a node or an edge, and a target attribute $\mathbf{y}_t$. For an evidence-discovery workflow, the objective is defined by a research query q and a target evidence set $\mathcal{Y}_q$.

Step 2: Constructing the task-relevant graph view. For task-specific models, RESEARCHARCADE retrieves a fixed multi-hop neighborhood $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ around the target entity. Its one-hop neighborhood is

$$\mathcal{N}_t^{(1)} = \{v \in \mathcal{V} \mid (t, v) \in \mathcal{E}\},$$

with the endpoints used when t is an edge. Higher-order neighborhoods are constructed recursively from the preceding hop. A task-specific model then predicts

$$\hat{\mathbf{y}}_t = f_\theta(\mathcal{G}_t).$$

The task is predictive when $\mathbf{y}_t$ is a discrete or ranked outcome and generative when the output is open-ended.

Generative tasks are trained using supervised fine-tuning with the loss

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\frac{1}{\sum_{t=1}^{T} L_t} \sum_{t=1}^{T} \sum_{i=1}^{L_t} \log p_\theta(y_{t,i} \mid y_{t,<i}, \mathcal{G}_t)\,, \tag{1}$$

where L_t is the length of the target output $\mathbf{y}_t$.

Evidence-discovery workflows may instead expose the graph through retrieval and traversal operations. Let $\mathcal{G}^{\leq \delta_q}$ denote the academic graph containing only information available by the temporal cutoff δ_q. At interaction step i, an agent selects an action according to

$$a_i \sim \pi_\theta(\cdot \mid q, a_{<i}, o_{<i})\,, \qquad o_i = \mathcal{T}\big(\mathcal{G}^{\leq \delta_q}, a_i\big)\,,$$

where $\mathcal{T}$ is the shared search and graph-access interface and o_i is the returned observation. After at most L interactions, the system produces a ranked evidence set

$$\hat{\mathcal{Y}}_q = g_\theta(q, a_{1:L}, o_{1:L}).$$

A fixed retrieval pipeline follows a predetermined sequence of retrieval and graph-expansion operations, whereas an interactive agent may adapt subsequent actions to previously observed evidence.

We instantiate this interface with six academic tasks and two AutoResearch evidence-discovery workflows. The six tasks are summarized in Table 1; the evidence-discovery workflows are defined below.

4.3 ACADEMIC TASK INSTANTIATIONS

We instantiate six academic tasks spanning scientific writing, literature use, peer review, and manuscript revision. Together, they evaluate whether the common graph interface can support predictive and generative models with different input and output requirements. The task definitions are summarized in Table 1, with additional details provided in Appendix A.3.

4.4 AUTORESEARCH EVIDENCE-DISCOVERY WORKFLOWS

Beyond task-specific prediction and generation, AutoResearch systems must discover sets of relevant evidence from a large and continuously evolving literature. We define two workflows that evaluate this capability at different levels of context. Both workflows are evaluated using fixed retrieval pipelines and interactive research agents under the same temporal and candidate-access constraints.

Table 1: **Summary of six academic task instantiations supported by RESEARCHARCADE.** Abbreviations: "or" denotes OpenReview, "ar" denotes ArXiv, CE denotes cross-entropy loss, and BCE denotes binary cross-entropy loss.

Task	Target Entity (Step 1)	Neighborhood (Step 2)	Loss	Type
Citation Prediction	ar_citation edge: content of the citing paragraph	ar_section nodes, ar_paragraph nodes, ar_figure nodes, ar_table nodes, ar_paper nodes	CE	Predictive
Paragraph Generation	ar_paragraph node: Textual content of the paragraph	ar_paragraph nodes, ar_table nodes, ar_figure nodes, ar_citation edges	SFT	Generative
Revision Retrieval	or_revision node: Index list of modified paragraphs	or_paragraph nodes from the original paper, or_review nodes	InfoNCE	Predictive
Revision Generation	or_paragraph node: Textual content of the revised paragraph	or_paragraph node of the original paper, or_review nodes	SFT	Generative
Acceptance Prediction	or_paper node: Paper decision	or_paper nodes, ar_paper nodes, ar_paragraph nodes, ar_figure nodes, ar_table nodes	BCE	Predictive
Rebuttal Generation	or_review node: Textual content of the author's response	or_review node of the official review being replied to, ar_paper node, ar_paragraph nodes, ar_figure nodes, ar_table nodes	SFT	Generative

4.4.1 CITATION SET COMPLETION

Citation Set Completion models the process of identifying the remaining evidence needed to support a local scientific statement. Each instance is constructed from a real paragraph containing between two and five resolved ArXiv citations. The input consists of the paragraph with all citation markers replaced by a neutral token, together with one revealed citation as an anchor. The target evidence set $\mathcal{Y}_q$ contains the remaining citations from the same paragraph.

Graph-based systems construct candidates by traversing the citation neighborhood of the revealed anchor, optionally combining graph candidates with lexical retrieval. The temporal cutoff δ_q is the initial submission date of the source paper, and only papers available by that date may be traversed or returned. To prevent direct answer leakage, the source paper, its paragraphs, and all relations originating from them are excluded from graph access.

In the fixed-retrieval setting, the system applies a predetermined retrieval and graph-expansion procedure. In the agentic setting, a model may iteratively search the corpus, traverse graph connections, inspect retrieved papers, and assemble its final evidence set. We evaluate ranked target recall at multiple cutoffs, MRR, complete-set recall, and set-level F1.

4.4.2 LITERATURE SET EXPANSION

Literature Set Expansion models automated literature snowballing from an initial research description. Given only the title and abstract of a source paper, with citation markers removed, the system returns a ranked set of earlier papers relevant to the source. The target set $\mathcal{Y}_q$ consists of the source paper's resolved references that were available before its initial submission.

Because an author's bibliography is selective rather than exhaustive, these references provide reproducible but incomplete relevance labels. Absolute recall therefore provides a lower bound on topical coverage, while paired comparisons use the same target set for every retrieval condition.

Candidate papers are obtained through lexical retrieval, expansion over the citation graph, or a combination of both. Graph expansion begins from lexically retrieved seed papers and follows only relations available before the source paper's submission date. The source paper and its associated paragraphs are excluded so that its own bibliography and citation edges cannot reveal the targets.

The fixed-retrieval setting applies a predetermined expansion procedure to produce a ranked list. In the agentic setting, a research agent may iteratively search, follow citation connections, and refine the selected literature set based on previous observations. Performance is measured using Recall, nDCG, and MAP at fixed retrieval budgets, with Recall@50 as the primary measure of evidence-set coverage.

Table 2: **Evaluation results across six academic tasks.** Each base model follows (Backbone, Training, Hop), where Backbone is the specific model, Training is Fixed or Trained, and #-hop is the number of hops of neighbors that a model can observe. (0-hop indicates no neighbors are observed.) The near-saturated LLM-judge score for revision generation reflects the judge's preference for low-edit outputs and is analyzed in Appendix A.9.4.

Citation Prediction				**Paragraph Generation**			
Model\Metric	Accuracy	AUC - ROC	MCC	Model\Metric	SBERT	Rouge-L	GPT-4o-mini
EMB (Longformer, Fixed, 1-hop)	0.970	0.427	0.050	Graph-LM (Qwen3-0.6B, Trained, 0-hop)	0.581	0.163	0.009
GNN (HANConv, Trained, 1-hop)	**0.989**	**0.995**	0.396	Graph-LM (Qwen3-0.6B, Trained, 1-hop)	0.624	**0.167**	0.244
GNN (HANConv, Trained, 3-hop)	0.987	0.993	**0.705**	Graph-LM (Qwen3-0.6B, Trained, 3-hop)	0.638	0.166	0.404
GNN (HANConv, Trained, 5-hop)	**0.989**	0.993	**0.705**	Graph-LM (Qwen3-0.6B, Trained, 5-hop)	**0.642**	0.165	0.344
Revision Retrieval				**Acceptance Prediction**			
Model\Metric	Precision@5	Recall@5	F-1@5	Model\Metric	Accuracy	AUC - ROC	MCC
EMB (Longformer, Fixed, 1-hop)	0.183	0.154	0.145	MLP (Linear, Trained, 1-hop)	0.513	0.479	0.025
GNN (HANConv, Trained, 1-hop)	**0.307**	0.325	**0.265**	GNN (HANConv, Trained, 1-hop)	0.507	0.465	0.000
GNN (HANConv, Trained, 3-hop)	**0.307**	0.324	**0.265**	GNN (HANConv, Trained, 3-hop)	**0.550**	**0.526**	**0.115**
Graph-LM (Qwen3-0.6B, Trained, 1-hop)	0.304	0.325	0.264	Graph-LM (Qwen3-0.6B, Trained, 1-hop)	0.470	0.478	-0.063
Graph-LM (Qwen3-0.6B, Trained, 3-hop)	0.306	**0.326**	**0.265**	Graph-LM (Qwen3-0.6B, Trained, 3-hop)	0.527	0.524	0.052
Revision Generation				**Rebuttal Generation**			
Model\Metric	SBERT	Rouge-L	GPT-4o-mini	Model\Metric	SBERT	Rouge-L	GPT-4o-mini
LLM (Qwen3-0.6B, Fixed, 1-hop)	0.321	0.210	0.447	LLM (Qwen3-0.6B, Fixed, 1-hop)	0.604	0.125	0.011
LLM (Qwen3-0.6B, Trained, 1-hop)	**0.733**	**0.554**	0.572	LLM (Qwen3-0.6B, Trained, 1-hop)	0.638	0.131	0.022
LLM (Qwen3-8B, Fixed, 1-hop)	0.704	0.446	0.889	LLM (Qwen3-8B, Fixed, 1-hop)	0.700	**0.154**	0.208
LLM (GPTOSS-120B, Fixed, 1-hop)	0.669	0.265	**0.999**	LLM (GPTOSS-120B, Fixed, 1-hop)	**0.703**	0.152	**0.884**

5 EXPERIMENTS

5.1 EXPERIMENT SETUP

Dataset: For the six academic task instantiations, we conduct experiments based on a subset of data in RESEARCHARCADE. For data from ArXiv, we mainly focus on papers in the Computer Science field and published within the last two years. For data collected from OpenReview, we primarily focus on the ICLR conferences within the past five years. Further detailed data usage information is provided in Appendix A.6.

Models and metrics: Across the six academic task instantiations, we evaluate embedding models, graph neural networks, large language models, and a graph-augmented language model (Graph-LM) adapted from Graph World Model (**?**), with task-appropriate predictive and generative metrics. Graph-LM aggregates multi-hop neighborhood features into graph tokens and prepends them to a Qwen3-0.6B backbone. Text and visual inputs are encoded with Longformer, Qwen3-Embedding-0.6B, and LLaVA-1.5-7B where applicable. Complete model configurations, evaluation metrics, and prompts are provided in Appendix A.6 and Appendix A.9.

AutoResearch Evidence-Discovery Workflows: We additionally evaluate *Citation Set Completion* and *Literature Set Expansion* under both fixed-retrieval and agentic settings. The fixed-retrieval evaluation uses 5,000 Citation Set Completion instances and 909 Literature Set Expansion instances. Agent evaluations use 150 frozen instances for each workflow. For every instance, the accessible graph is restricted to information available by the source paper's initial submission date. The source paper and its associated paragraphs are excluded from traversal so that source-generated citation edges cannot reveal the target evidence. For Literature Set Expansion, the source paper's resolved references serve as reproducible but incomplete relevance labels; we therefore interpret paired differences between conditions rather than absolute recall as the primary evidence.

Retrieval Conditions: We compare lexical retrieval using BM25 with graph-augmented retrieval using either the correct academic graph or a degree-preserving rewired graph. Hybrid systems combine the lexical and graph rankings while preserving the same candidate budget. Correct and rewired conditions use identical nodes, temporal filters, retrieval procedures, and budgets, differing only in graph connectivity.

Agent Conditions: Interactive agents access the same lexical-search and graph-traversal operations under fixed tool-call and observation budgets. To ensure that the graph intervention reaches the agent, we use a controlled interaction protocol with a disclosed minimum research phase before final submission. Correct and rewired conditions use byte-identical prompts, tool schemas, and budgets, differing only in graph connectivity. The correct-connectivity and rewired-connectivity

Table 3: **Graph-augmented retrieval on the two AutoResearch workflows and existing structural controls.** Results use 5,000 Citation Set Completion (CSC) instances and 909 Literature Set Expansion (LSE) instances. Δ compares the correct and degree-preserving rewired hybrids.

Condition	CSC Recall@20	LSE Recall@50
BM25	0.3516	0.1913
Rewired hybrid	0.3396	0.2516
Correct hybrid	**0.6303**	**0.4266**
Δ (95% CI)	$+\mathbf{0.2907}$ $[+0.2789, +0.3026]$	$+\mathbf{0.1750}$ $[+0.1644, +0.1858]$
Random anchor	0.023	–
Degree-matched anchor	0.047	–
Anchor text only	0.247	–
Correct graph only	0.547	–
Citation edges only	–	0.401
Paragraph edges only	–	0.180

agents use identical model prompts and tools, with only the underlying graph changed. We evaluate Qwen2.5-7B and Qwen3-8B on both workflows and additionally evaluate Qwen2.5-32B on Citation Set Completion.

Evidence-Discovery Metrics: Citation Set Completion is primarily evaluated using Recall@20 for fixed retrieval and Recall@10 for agents. Literature Set Expansion is primarily evaluated using Recall@50. We also report MRR, nDCG, MAP, and set-level metrics where applicable. Statistical comparisons use paired bootstrap confidence intervals with 10,000 resamples and paired permutation tests, with Holm correction within each confirmatory family.

5.2 Task-Specific Academic Results

Table 2 demonstrates that the common interface supports predictive and generative tasks across embedding models, graph neural networks, language models, and the Graph-LM. These experiments are intended to evaluate interface breadth rather than to compare performance across tasks that have different inputs, outputs, and metrics. Richer graph neighborhoods and multi-modal inputs provide task-dependent benefits. Complete task-by-task analyses, ablations, and case studies are reported in Appendix A.9.

5.3 Graph Structure Improves Evidence Discovery

Table 3 reports the fixed-retrieval results. For Citation Set Completion, the correct hybrid improves Recall@20 from 0.3516 with BM25 to 0.6303. For Literature Set Expansion, it improves Recall@50 from 0.1913 to 0.4266. More importantly, the correct graph outperforms its degree-preserving rewired counterpart by 0.2907 and 0.1750, respectively, with paired confidence intervals excluding zero. Because the two graph conditions share the same nodes, degree distribution, retrieval procedure, and candidate budget, these differences show that the gains depend on task-relevant citation connectivity rather than merely adding more retrieval capacity.

The anchor controls show that the Citation Set Completion gain cannot be explained by anchor degree or popularity: replacing the true anchor with a degree-matched paper reduces Recall@20 from 0.547 to 0.047. In Literature Set Expansion, citation edges alone reach 0.401 Recall@50, compared with 0.180 for paragraph edges alone, identifying citation connectivity as the primary structural source of the gain.

5.4 Graph Connectivity Improves Evidence Access for Interactive Agents

The correct-versus-rewired intervention remains effective when graph access is placed inside an interactive agent loop. As shown in Table 4, correct connectivity improves evidence recall across both workflows and all evaluated model scales. On Citation Set Completion, the gain ranges from 0.0889 to 0.1028 Recall@10 across models. On Literature Set Expansion, the gain is 0.0660 and 0.0644

Table 4: **Graph-intervention and matched-pool agent results.** Correct − Rewired is reported for all models. Agent − Replay and absolute baselines are reported for Qwen3-8B. All rows use the same 150 frozen instances within each workflow.

Workflow	Model	Metric	Correct − Rewired (95% CI)	Agent − Replay (95% CI)	BM25	Flat	Best Agent	Fixed Hybrid
Citation Set Completion	Qwen2.5-7B	Recall@10	$+\mathbf{0.0889}$ $[+0.0428, +0.1378]$	–	–	–	–	–
	Qwen3-8B	Recall@10	$+\mathbf{0.1017}$ $[+0.0528, +0.1522]$	$+0.0072$ $[-0.0317, +0.0456]$	0.236	0.254	0.426	**0.427**
	Qwen2.5-32B (Int4)	Recall@10	$+\mathbf{0.1028}$ $[+0.0550, +0.1533]$	–	–	–	–	–
Literature Set Expansion	Qwen2.5-7B	Recall@50	$+\mathbf{0.0660}$ $[+0.0387, +0.0934]$	–	–	–	–	–
	Qwen3-8B	Recall@50	$+\mathbf{0.0644}$ $[+0.0462, +0.0843]$	-0.0046 $[-0.0119, -0.0002]$	0.176	0.123	0.281	**0.423**

Recall@50 for Qwen2.5-7B and Qwen3-8B, respectively. Every paired confidence interval excludes zero, showing that useful graph structure continues to affect the evidence reached by interactive research agents.

5.5 Graph Gains Primarily Come from Evidence Access

The interaction traces identify where these gains arise. In Citation Set Completion, 31.2% of targets observed by the correct-connectivity agent are graph-only targets that the flat agent never observes, compared with 8.1% under rewired connectivity. At least one such target is surfaced in 43.3% of correct-graph episodes, versus 12.7% of rewired episodes. Thus, correct connectivity substantially changes the evidence available to the agent. Consistently, the dominant failure, target never retrieved, falls from 61.3% to 31.3%, while observed-but-not-selected failures rise from 10.0% to 26.7%. Correct connectivity therefore shifts the bottleneck from evidence acquisition toward final evidence selection.

However, under the evaluated models and protocol, no agent significantly outperforms its matched replay after Holm correction. The primary value of the graph therefore lies in assembling task-relevant evidence; the agents do not reliably add further gains through sequential planning over the candidates they observe.

The contrast is especially clear for Literature Set Expansion: the flat search agent reaches 0.123 Recall@50, below the 0.176 BM25 one-shot baseline, while the fixed correct hybrid exceeds the best interactive agent by 0.142. Agent interaction alone therefore does not explain the graph gain.

6 Conclusion

We introduced ResearchArcade, a graph interface that unifies multi-source, multi-modal, and temporally evolving academic data for task-specific models and interactive AutoResearch workflows. Six academic task instantiations demonstrate the breadth of the interface across predictive and generative settings. Two evidence-discovery workflows further show that correct graph connectivity consistently expands the evidence accessible to AutoResearch systems in both fixed and interactive retrieval. By combining degree-preserving rewiring with matched-pool replay, we further localize the measured gains to evidence access: sequential interaction does not improve over the candidate pools exposed by the tools. Thus, ResearchArcade provides not only a reusable interface for academic tasks, but also a controlled environment for identifying which components of AutoResearch systems contribute to their performance.

Reproducibility Statement

The anonymous repository provides the data-construction and evaluation code, frozen task splits, prompts, tool schemas, graph hashes, per-episode agent trajectories, and commands used to reproduce the reported tables. Dataset construction and processing are documented in Appendix A.2, model and task configurations in Appendix A.6, and the complete evidence-discovery protocol and paired analyses in Appendix A.4. The AutoResearch results are keyed by model, condition, and frozen instance identifier; correct and rewired conditions share prompts, budgets, and temporal filters and differ only in graph connectivity.

AI Usage Statement

We used generative AI tools, including OpenAI ChatGPT/Codex and Anthropic Claude/Claude Code, to assist with literature discovery, feedback on experimental design, implementation and debugging of data-processing and evaluation scripts, inspection and summarization of experimental outputs, and drafting, editing, and formatting the manuscript and tables. LLMs were also used as research components where explicitly described in the paper, including as evaluated models, LLM-based judges, and generators of derived dataset fields marked as LLM-generated in the schema. The authors reviewed and tested all AI-assisted code, checked reported results against saved outputs and evaluation artifacts, verified the cited literature and AI-assisted text, and made all final methodological and interpretive decisions. We take responsibility for the complete content of this work, including all text, claims, code, and artifacts produced with AI assistance.

References

Mike D'Arcy, Alexis Ross, Erin Bransom, Bailey Kuehl, Jonathan Bragg, Tom Hope, and Doug Downey. Aries: A corpus of scientific paper edits made in response to peer reviews. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

Tao Feng, Chuanyang Jin, Jingyu Liu, Kunlun Zhu, Haoqin Tu, Zirui Cheng, Guanyu Lin, and Jiaxuan You. How far are we from AGI: are llms all we need? *Trans. Mach. Learn. Res.*, 2024.

Sujatha Das Gollapalli and Xiaoli Li. EMNLP versus ACL: Analyzing NLP research over time. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2002–2006, Lisbon, Portugal, 2015. Association for Computational Linguistics.

Léane Jourdan, Florian Boudin, Richard Dufour, Nicolas Hernandez, and Akiko Aizawa. Pararev: Building a dataset for scientific paragraph revision annotated with revision instruction. In *Proceedings of the First Workshop on Writing Aids at the Crossroads of AI, Cognitive Science and NLP (WRAICOGS 2025)*, 2025.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine Van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. A dataset of peer reviews (peerread): Collection, insights and nlp applications. *CoRR*, 2018.

Ilia Kuznetsov, Jan Buchmann, Max Eichler, and Iryna Gurevych. Revise and resubmit: An intertextual model of text-based collaboration in peer review. *Computational Linguistics*, 2022.

Shengzhi Li and Nima Tajbakhsh. Scigraphqa: A large-scale synthetic multi-turn question-answering dataset for scientific graphs. *CoRR*, 2023.

Guanyu Lin, Tao Feng, Pengrui Han, Ge Liu, and Jiaxuan You. Paper copilot: A self-evolving and efficient LLM system for personalized academic assistance. *CoRR*, 2024.

Yaowenqi Liu, Bingxu Meng, Rui Pan, Yuxing Liu, Jerry Huang, Jiaxuan You, and Tong Zhang. GUIDE: Towards scalable advising for research ideas. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42813–42834, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1984. URL https://aclanthology.org/2026.acl-long.1984/.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S Weld. S2orc: The semantic scholar open research corpus. *CoRR*, 2019.

Tarek Saier, Johan Krause, and Michael Färber. unarxive 2022: All arxiv publications pre-processed for nlp, including structured full-text and citation network. In *2023 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pp. 66–70. IEEE, 2023.

Yang Tian, Zheng Lu, Mingqi Gao, Zheng Liu, and Bo Zhao. Mmcr: Benchmarking cross-source reasoning in scientific papers. *CoRR*, 2025.

Yunpei Tian, Gang Li, and Jin Mao. Predicting the evolution of scientific communities by interpretable machine learning approaches. *Journal of Informetrics*, 17(2):101399, 2023.

Renqiu Xia, Song Mao, Xiangchao Yan, Hongbin Zhou, Bo Zhang, Haoyang Peng, Jiahao Pi, Daocheng Fu, Wenjie Wu, Hancheng Ye, et al. Docgenome: An open large-scale scientific document benchmark for training and testing multi-modal large language models. *CoRR*, 2024.

Fanjin Zhang, Shijie Shi, Yifan Zhu, Bo Chen, Yukuo Cen, Jifan Yu, Yelin Chen, Lulu Wang, Qingfei Zhao, Yuqing Cheng, et al. Oag-bench: a human-curated benchmark for academic graph mining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6214–6225, 2024.

A APPENDIX

A.1 DATA DESCRIPTION IN RESEARCHARCADE

The detailed description of RESEARCHARCADE is shown in Figure 3.

The statistical overview of data collected from ArXiv is illustrated in Table 5.

The statistical overview of data collected from OpenReview is illustrated separately in Table 6 (ICLR), Table 7 (NeurIPS), Table 8 (ICML), and Table 9 (EMNLP).

The statistical overview of openreview_arxiv table is shown in Table 10 (ICLR), Table 11 (NeurIPS), Table 12 (ICML), and Table 13 (EMNLP).

Table 5: **Statistic overview of the data collected from ArXiv by primary category.** Note that some papers may belong to multiple subdomains within a category. Counts are aggregated by top-level category prefix.

Category	**#papers**	**#sections**	**#paragraphs**	**#figures**	**#tables**	**#authors**
cs	57357	849448	11306247	1309302	528055	14385
stat	9938	106751	1879176	177750	47557	613
physics	8421	69334	1013456	122287	18395	349
math	5263	50397	1293357	71226	14666	766
eess	5069	37376	479536	54377	19070	1511
q-bio	2797	24588	353634	34342	10722	220
cond-mat	1991	16794	256023	22744	2905	81
astro-ph	690	6858	97977	10352	1962	47
nlin	388	3009	49485	4559	429	17
econ	300	2683	63133	3920	1371	30
q-fin	253	2468	42267	5360	1287	45
Total	66918	569501	8014095	876636	324648	14391

Table 6: **Statistic overview of the data collected from ICLR conferences, sourced from the OpenReview.** Note that no ICLR conference was held in 2015 and 2016. Additionally, revisions of submissions from the ICLR 2013 and 2014 conferences are not accessible on the OpenReview.

Year	**#papers**	**#authors**	**#reviews**	**#paragraphs**	**#revisions**	**#papers_authors**	**#papers_reviews**	**#papers_revisions**	**#reviews_revisions**
2025	8701	27742	190934	1526799	13989	42541	190934	13989	97051
2024	5750	18077	99525	389973	1251	25297	99520	1251	11971
2023	3793	11819	55301	893211	9445	15742	55301	9445	39871
2022	2617	8155	39750	614294	6508	10505	39750	6508	28321
2021	2594	7661	32113	566963	6593	9782	32113	6593	22786
2020	2213	6963	21132	556021	6878	9117	21132	6878	14773
2019	1419	4387	16620	306915	3671	5618	16620	3671	11503
2018	935	2820	9164	352761	4929	3512	9164	4929	8374
2017	490	606	6988	104648	1203	869	6988	1203	4206
2014	69	65	548	2803	/	84	548	/	/
2013	67	56	373	2691	/	74	373	/	/
Total	28648	88351	472448	5317079	54467	123141	472443	54467	238856

Table 7: **Statistic overview of the data collected from NeurIPS conferences, sourced from the OpenReview.** Note that no NeurIPS conference data before 2021 is available on OpenReview. Additionally, no revision is allowed during the rebuttal process.

Year	#papers	#authors	#reviews	#paragraphs	#papers_authors	#papers_reviews
2025	5529	21687	125739	370664	30352	125739
2024	4236	15430	75588	279708	21673	75588
2023	3394	11191	64528	218856	15863	64528
2022	2824	9102	46915	180202	12561	46915
2021	2768	7600	37952	167213	11744	37952
Total	18751	65010	350722	1216643	92193	350722

Table 8: **Statistic overview of the data collected from ICML conferences, sourced from the OpenReview.** Note that no ICML conference data before 2023 is available on OpenReview. Additionally, peer review data is only available for the year 2025, no revision is allowed during the rebuttal process, and no extracted paragraph data because the ICML PDFs are double-column-structured.

Year	#papers	#authors	#reviews	#papers_authors	#papers_reviews
2025	3422	13279	38974	17871	38974
2024	2610	9516	/	13050	/
2023	1828	6186	/	8121	/
Total	7860	28981	38974	39042	38974

Table 9: **Statistic overview of the data collected from EMNLP conferences, sourced from the OpenReview.** Note that only the EMNLP conference in 2023 is available on OpenReview. Additionally, no revision is allowed during the rebuttal process.

Year	#papers	#authors	#reviews	#paragraphs	#papers_authors	#papers_reviews
2023	2019	6696	22731	118184	10015	22731

Table 10: **Statistic overview of** openreview_arxiv **table on ICLR.** Note that no ICLR conference was held in 2015 and 2016. Additionally, revisions of submissions from the ICLR 2013 and 2014 conferences are not accessible on the OpenReview.

Year	2025	2024	2023	2022	2021	2020	2019	2018	2017	2014	2013
#openreview_arxiv	3077	2033	1469	1050	1068	866	583	424	248	53	50

Table 11: **Statistic overview of** openreview_arxiv **table on NeurIPS.** Note that no NeurIPS conference before 2021 is available on OpenReview.

Year	2025	2024	2023	2022	2021
#openreview_arxiv	2898	2303	1808	1430	1368

Table 12: **Statistic overview of** openreview_arxiv **table on ICML.** Note that no ICML conference before 2023 is available on OpenReview.

Year	2025	2024	2023
#openreview_arxiv	1828	1398	998

Table 13: **Statistic overview of** openreview_arxiv **table on EMNLP.** Note that only the EMNLP conference in 2023 is available on OpenReview.

Year	2023
#openreview_arxiv	1017

A.2 Data Construction Procedure in ResearchArcade

In this section, we detail the data collection, filtering, and compilation workflows implemented in ResearchArcade. The ArXiv data collection process is provided in A.2.1, the continuous crawling process is described in A.2.2, and the OpenReview data collection process is outlined in A.2.3.

To assess data quality, we manually inspected 300 randomly sampled papers for paragraph–section linkage, OpenReview–ArXiv alignment, paper–author relations, and record completeness. We found no errors among the inspected fields; this spot check should not be interpreted as a corpus-wide precision estimate. The complete source code and tested data splits are provided in the anonymous supplementary repository.

A.2.1 ArXiv Data Construction

We develop a multi-stage pipeline to collect and structure papers from ArXiv. The process begins by selecting target papers using ArXiv IDs or publication date ranges. Using the ArXiv API, we download the LaTeX source files together with basic metadata. The sources are then processed through a seven-stage pipeline that converts raw LaTeX into structured graph representations.

Stage 1: Paper Source File Downloading For each paper, we download and unpack the LaTeX source archive into a working directory. We also collect metadata including author names, paper categories, submission dates, paper version, and abstracts.

Stage 2: Author Information Processing Author names are first obtained from ArXiv metadata, which provides plain-text names without persistent identifiers. To enrich this information, we query the Semantic Scholar API using the ArXiv ID. We extract author identifiers, names, and optional auxiliary information (e.g., homepages). Authors are stored in a dedicated table, and paper–author relationships are recorded with sequence numbers to preserve author order.

Stage 3: Section-Level Decomposition We identify paper sections by recursively traversing the LaTeX structure. The system assumes a three-level hierarchy (`section`, `subsection`, `subsubsection`). When sectioning commands (e.g., `\section...`, `\subsection...`, `\subsubsection...`) are detected, we extract section titles, store section content, and record their positions within the paper.

Stage 4: Paragraph-Level Information Extraction Within each section, text is segmented into paragraphs based on blank lines, explicit line breaks, and environment boundaries. Content belonging to figures, tables, or display-math environments is excluded to preserve clean textual paragraphs. Each paragraph is assigned paper-level and section-level ordering. We detect citation commands (e.g., `\cite...`) and extract citation keys and titles. References to figures and tables are identified through cross-reference commands (e.g., `\ref...`). These links are stored in relational tables that connect paragraphs to citations, figures, and tables.

Stage 5: Citation Information Processing Citation metadata is collected from BibTeX files and compiled bibliography environments (e.g., `.bbl` files or `\beginthebibliography` blocks). We extract citation keys, titles, authors, and venue information, and detect ArXiv identifiers when present. For references without explicit ArXiv IDs, we apply a two-step resolution process. First, we query Semantic Scholar to retrieve reference lists with external identifiers. Second, we align these references with local entries using normalized title similarity with a conservative threshold. Unmatched references are retained in the database without ArXiv IDs.

Stage 6: Figure and Table Extraction Figures and tables are detected and indexed using their environment labels and captions at both document and section levels. Each figure or table is stored as a structured object linked to its parent paper and, when applicable, its enclosing section. At the paragraph level, cross-references to figures and tables are resolved using a global label index. Only figures and tables that are referenced in the text are retained, ensuring semantic grounding. This enables fine-grained links between paragraphs and the visual or tabular content they describe.

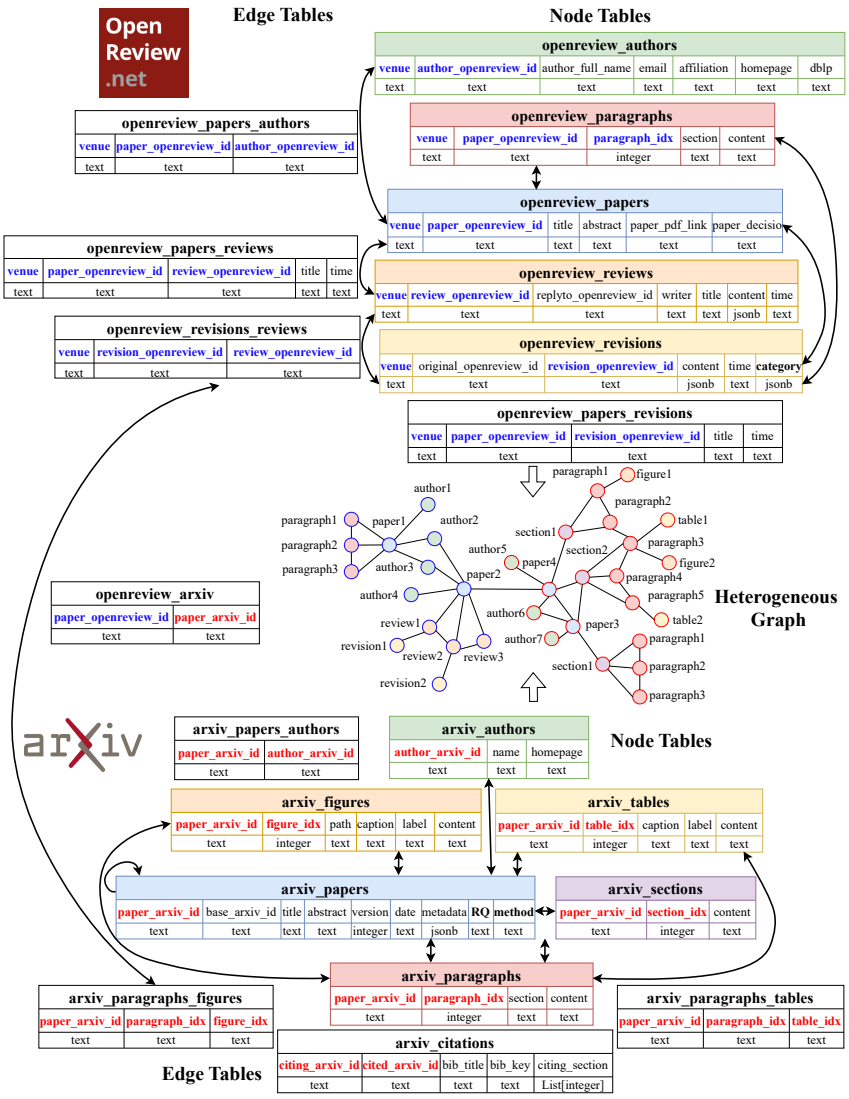


Figure 3: **A comprehensive overview of RESEARCHARCADE.** RESEARCHARCADE uses a multi-table format with graph structures to collect data from different sources with multiple modalities. Tables are classified into node tables (colored) or edge tables (black and white). The blue (denoting the OpenReview part) or red (denoting the ArXiv part) columns represent the unique identification of each node or edge, and the remaining columns represent the features of the nodes or edges. Author metadata contain email domains rather than email addresses. The black bold columns are generated by LLM. The conversion from the multiple tables to heterogeneous graphs is straightforward.

Stage 7: Graph Construction and Storage All extracted elements are organized into a heterogeneous graph. Nodes represent papers, sections, paragraphs, figures, tables, and authors. Edges encode relationships such as paper–section, section–paragraph, paragraph–citation, paragraph–figure, paragraph–table, paper–author, paper–paper citations, and paper–category links. Citation edges are deduplicated and store both original bibliographic metadata and resolved ArXiv identifiers. This graph serves as the foundation for downstream tasks.

A.2.2 CONTINUOUS CRAWLING FROM ARXIV

RESEARCHARCADE supports continuous and automated data acquisition through a fault-tolerant crawling and processing pipeline. The pipeline design reflects the multi-stage extraction procedure

described above and enables the dataset to remain synchronized with newly released ArXiv papers while preserving consistency of the existing heterogeneous graph.

The major stages of the pipeline are as follows:

1. **ArXiv Identifier Retrieval.** A scheduled job (e.g., daily or weekly) queries the ArXiv API to collect newly published paper identifiers within a predefined time window, supporting both incremental updates and recovery from transient failures.
2. **Source Archive Downloading.** For each new ArXiv ID, the compressed LaTex source archive is automatically downloaded and unpacked into a staging directory to support parsing and processing.
3. **Incremental Graph Construction and Update.** Each paper is processed using the same pipeline described in Appendix A.2.1, converting source files into structured graph entities, including papers, sections, paragraphs, figures, tables, citations, and their relations.
4. **External Metadata Enrichment.** Additional metadata are added using external services, including author enrichment via the Semantic Scholar API and citation resolution through title and ArXiv ID based matching.

This automated pipeline allows RESEARCHARCADE to evolve continuously by integrating newly published content in a consistent and reproducible way, supporting longitudinal analysis of academic structures and relationships.

The pipeline also supports two interchangeable storage backends: a PostgreSQL database and a CSV-based file store. PostgreSQL provides efficient querying and scalability for large-scale experiments, while the CSV backend offers a lightweight, portable option that requires no database deployment and can be easily used across different computing environments.

A.2.3 OPENREVIEW DATA CONSTRUCTION

The detailed procedures used to collect, filter and compile data from OpenReview into RESEARCHARCADE are as follows.

Firstly, by providing a conference ID, we utilize the OpenReview API to retrieve the authors' IDs, titles, abstracts, decisions, PDF links, and unique submission IDs for each paper presented at the conference. We excluded the withdrawn papers and the papers without author IDs. This step mainly contributes to the construction of the $\mathrm{or_papers}$ table and the $\mathrm{or_papers_authors}$ table.

Given the author IDs, the OpenReview API returns detailed author metadata, including full name, email domain, institutional affiliation, homepage URL, and DBLP entry. Note that, for some authors, the homepage and DBLP fields are missing from the metadata. These records constitute the $\mathrm{authors}$ table. Moreover, each author in OpenReview has a unique author ID, although they might have the same name.

The OpenReview API also provides access to official reviews and comments associated with each paper submission. For each review, we retrieve its ID, the ID of the review it responds to, and its timestamp. It is important to note that the official review, meta-review, and paper decision directly reply to the submission ID. We further ensure that each review comment includes reviewer scores and validated content like contribution summary, weaknesses assessment, etc. The filtered data is then used to form the $\mathrm{or_reviews}$ table and the $\mathrm{or_papers_reviews}$ table.

To construct the $\mathrm{or_paragraphs}$ table, we first download the PDF files and utilize $\mathrm{pdfminer}$ to extract the text from papers. The extracted text is then organized into paragraphs with unique $\mathrm{paragraph_idx}$ within each paper based on the distance between consecutive words. To improve the quality of the extracted paragraphs, we exclude the references and appendices of each paper, as these sections in PDF files often contain dense formatting (e.g., multi-column layouts, tables, equations, and citation lists) that can degrade text extraction and introduce noise, leading to fragmented or duplicated paragraphs.

For the $\mathrm{or_revisions}$ table and the $\mathrm{or_papers_revisions}$ table, we focus on the content of the revisions, since it directly captures the substantive changes made during iterative editing, whereas original paragraphs contribute little semantic signal and may introduce spurious correlations. Therefore, we begin by downloading the PDFs of all the revised papers and storing their content in the $\mathrm{or_paragraphs}$

table. Note that each revised paper also has a unique submission ID. Then, based on the timestamp of each revised submission, we identify pairs of original and revised papers and use difflib to extract the differences between the original and the revised texts. Finally, the locations of these differences are linked back to each paragraph.

Finally, to construct the or_revisions_reviews table, we assume that each revision is created as a result of discussions between the reviewers and the authors. And the specific discussions are identified if they occurred within the time period between the previous revision and the current revision. Thus, this table is constructed by leveraging the time information from the or_revisions table and or_reviews table.

A.3 Academic Task Definitions

In this section, we present concise definitions of the six proposed academic tasks using the two-step schema described in Section 4.2, including citation prediction (Section A.3.1), paragraph generation (Section A.3.2), revision retrieval (Section A.3.3), revision generation (Section A.3.4), acceptance prediction (Section A.3.5, and rebuttal generation (Section A.3.6).

A.3.1 Citation Prediction

Step 1: The target entity t is an $\mathrm{arxiv_paragraph_citation}$ edge, representing a citation made by a source paragraph. The label $\mathbf{y}_t$ denotes the index of the ground-truth cited paper that the target paragraph should reference.

Step 2: The academic graph $\mathcal{G}_t$ corresponds to the full paper containing the target paragraph. It includes $\mathrm{arxiv_section}$, $\mathrm{arxiv_paragraph}$, $\mathrm{arxiv_figure}$, and $\mathrm{arxiv_table}$ nodes. Paragraphs are sequentially linked. $\mathrm{arxiv_section}$ nodes provide the section title; $\mathrm{arxiv_citation}$ nodes represent external cited papers and are connected to citing paragraphs via $\mathrm{arxiv_paragraph_citation}$ edges.

A.3.2 Paragraph Generation

Step 1: The target entity t is an $\mathrm{arxiv_paragraph}$ node, with its textual content serving as the ground truth label $\mathbf{y}_t$.

Step 2: The academic graph $\mathcal{G}_t$ for this task includes the adjacent $\mathrm{arxiv_paragraph}$ nodes retrieved from the k-hop neighborhood (with k as a parameter), sequentially connected according to their order in the paper. Multi-modal nodes $\mathrm{arxiv_figure}$ and $\mathrm{arxiv_table}$ are also given, each linked to their corresponding paragraphs. $\mathrm{arxiv_citation}$ is added as external nodes connected to the citing paragraphs.

A.3.3 Revision Retrieval

Step 1: The target entity t in this task is an $\mathrm{openreview_revision}$ node, where the index list of the modified paragraphs in its attributes is the label $\mathbf{y}_t$ for this task.

Step 2: The academic graph $\mathcal{G}_t$ constructed in this task consists of two parts: First, the paragraphs from the original paper, with node type $\mathrm{openreview_paragraph}$, are retrieved from the 2-hop neighborhood, according to the $\mathrm{openreview_paper_revision}$ and the $\mathrm{openreview_paragraph}$ table. These paragraphs are sequentially connected based on their order; Second, the reviews, with node type $\mathrm{openreview_review}$, are also retrieved from the 2-hop neighborhood, according to the $\mathrm{openreview_paper_revision}$ and the $\mathrm{openreview_papers_review}$ table. They are connected based on their $\mathrm{review_openreview_id}$ and $\mathrm{replyto_openreview_id}$ attributes.

A.3.4 Revision Generation

Step 1: The target entity t in this task is a paragraph that has been revised. A revised paragraph is obtained based on the $\mathrm{revision_openreview_id}$ and the index list of the modified paragraphs for each $\mathrm{openreview_revision}$ node. The textual content of the revised paragraph is the label $\mathbf{y}_t$.

Step 2: To construct the academic graph $\mathcal{G}_t$ for this task, two types of nodes from t's neighborhood need to be retrieved: First, the corresponding paragraph from the original paper, with node type $\mathrm{openreview_paragraph}$, is retrieved from the 2-hop neighborhood based on the corresponding $\mathrm{openreview_revision}$ node and the $\mathrm{openreview_paragraph}$ table; Second, the reviews, with node type $\mathrm{openreview_review}$, are also retrieved from the 2-hop neighborhood based on the corresponding $\mathrm{openreview_revision}$ node, along with the $\mathrm{openreview_paper_revision}$ and the $\mathrm{openreview_papers_review}$ tables. These reviews are connected via their $\mathrm{review_openreview_id}$ and $\mathrm{replyto_openreview_id}$ attributes.

A.3.5 Acceptance Prediction

Step 1: Node $\mathrm{or_paper}$ is the target entity t in this task, and the paper's decision (Accept or Reject) is the label $\mathbf{y}_t$.

Step 2: The academic graph $\mathcal{G}_t$ is constructed using the data from ArXiv: First, relevant paragraphs, with node type $\mathrm{arxiv_paragraph}$, are retrieved from the 2-hop neighborhood, according to the $\mathrm{openreview_arxiv}$ and the $\mathrm{arxiv_paragraph}$ tables, with sequential connections reflecting their order. Second, the related figures, with node type $\mathrm{arxiv_figure}$, are retrieved through the $\mathrm{arxiv_paragraph_figure}$ table, with each figure connected to a specific paragraph. Finally, relevant tables, with node type $\mathrm{arxiv_table}$, are retrieved via the $\mathrm{arxiv_paragraph_table}$ table.

A.3.6 Rebuttal Generation

Step 1: The author's rebuttal response (can be inferred from the $\mathrm{openreview_review}$ node's title), with node type $\mathrm{openreview_review}$, is the target entity t in this task. The label $\mathbf{y}_t$ is the textual content of the response.

Step 2: The academic graph $\mathcal{G}_t$ is constructed as follows: Initially, the related official review, with node type $\mathrm{openreview_review}$, is retrieved based on the $\mathrm{replyto_openreview_id}$ attribute of t. Then, the corresponding paper graph is retrieved from ArXiv data using the same procedure as in Section A.3.5, which contains the relevant paragraphs with figures and tables.

A.4 AutoResearch Evidence-Discovery Workflows

This section provides construction, evaluation, and diagnostic details for Citation Set Completion and Literature Set Expansion. All reported results use frozen instances and temporally restricted graphs. The fixed-retrieval experiments contain 5,000 Citation Set Completion instances and 909 Literature Set Expansion instances; the agent experiments use 150 frozen instances from each workflow.

A.4.1 Instance Construction and Temporal Validity

For Citation Set Completion, each instance is derived from a paragraph with two to five resolved ArXiv citations. One citation is revealed as an anchor, and the remaining citations form the target evidence set. For Literature Set Expansion, the input is a source paper's title and abstract, and its resolved earlier references form the reproducible target set. Because this bibliography is selective and 68% of bibliography entries in the frozen agent set do not resolve to an in-corpus paper, absolute Literature Set Expansion recall is a lower bound; paired comparisons use the same targets across all conditions.

The 1,200 frozen Literature Set Expansion instances were partitioned before evaluation into a 291-instance development split and a 909-instance held-out screen split. We performed a grouped split over $(\text{year}, \text{primary ArXiv category})$ clusters, keeping every cluster within a single split. No instances were removed after freezing. All reported fixed-retrieval results use the 909-instance screen split, and all 150 agent instances are nested within it. Results on all 1,200 instances are directionally consistent and are retained as a robustness check.

For both workflows, graph access is restricted to information available by the source paper's initial submission date. Paper identifiers are normalized to version-stripped ArXiv identifiers before masking. The source paper, its paragraphs, and all source-originating edges are excluded from retrieval and traversal. The final audit found zero future-dated or source-derived results.

A.4.2 Retrieval and Agent Protocol

Fixed retrieval compares BM25, graph expansion, and reciprocal-rank-fused hybrids using either the correct citation graph or a degree-preserving rewired graph. Correct and rewired conditions share the same nodes, temporal filters, retrieval procedure, and candidate budget. Citation Set Completion expands at most two hops from the anchor using Adamic–Adar scores. Literature Set Expansion follows at most three hops from lexical seeds with a per-node cap of 25. Hybrid rankings use reciprocal rank fusion with $k = 60$ and return at most 50 candidates.

Agents use a controlled interaction protocol with a disclosed minimum research phase, at most eight tool calls, a 6,000-token observation budget, ten candidates per call, one format-repair turn, and one forced-answer turn on budget exhaustion. Correct and rewired agents receive byte-identical prompts and tool schemas; only the graph behind traversal differs. Successful tool uptake is 94–100% in all

Table 14: **Complete fixed-retrieval results for Citation Set Completion.** All conditions are evaluated on the same 5,000 frozen instances.

Condition	Recall@5	Recall@10	Recall@20	MRR	Set F1
BM25	0.212	0.276	0.352	0.223	0.076
Rewired graph	0.031	0.056	0.091	0.035	0.016
Correct graph	0.350	0.451	0.547	0.339	0.125
Rewired hybrid	0.185	0.258	0.340	0.167	0.070
Correct hybrid	**0.389**	**0.511**	**0.630**	**0.375**	**0.141**
Random anchor	0.009	0.015	0.023	0.009	0.004
Degree-matched anchor	0.016	0.029	0.047	0.017	0.008
Anchor text only	0.132	0.184	0.247	0.134	0.050

Table 15: **Complete fixed-retrieval results for Literature Set Expansion.** Results use the 909-instance held-out screen split.

Condition	Recall@20	Recall@50	nDCG@50
BM25	0.129	0.191	0.177
Rewired graph	0.096	0.153	0.124
Correct graph	0.283	0.401	**0.366**
Rewired hybrid	0.158	0.252	0.203
Correct hybrid	**0.288**	**0.427**	0.362
Citation edges only	0.283	0.401	**0.366**
Paragraph edges only	0.152	0.180	0.176

reported graph-agent conditions. Statistical comparisons use 10,000 paired bootstrap resamples and 10,000 paired sign-flip permutations, with Holm correction within each declared confirmatory family.

A.4.3 Complete Fixed-Retrieval Results

On the primary metrics, the correct hybrid exceeds the rewired hybrid by 0.2907 Recall@20 for Citation Set Completion (95% CI $[0.2789, 0.3026]$) and by 0.1750 Recall@50 for Literature Set Expansion (95% CI $[0.1644, 0.1858]$). A degree-matched but incorrect anchor reaches only 0.047 Recall@20 in Citation Set Completion, compared with 0.547 for the true anchor under the correct graph.

A.4.4 Agent Results and Evidence Access

Table 16 reports absolute performance for Qwen3-8B, for which all fixed and interactive conditions are available. The correct-versus-rewired gains replicate with Qwen2.5-7B on both workflows and with Qwen2.5-32B (Int4) on Citation Set Completion, as reported in Table 4.

For Qwen3-8B, correct connectivity reduces the dominant Citation Set Completion failure, target never retrieved, from 61.3% to 31.3%. At the same time, observed-but-not-selected failures increase from 10.0% to 26.7%, showing that final evidence selection becomes the remaining bottleneck.

A.4.5 Matched-Pool Analysis and Example Trace

No agent significantly outperforms a replay supplied with the same observed candidate pool after Holm correction. For Qwen3-8B, the correct-traversal agent minus its replay is $+0.0072$ in Citation Set Completion (95% CI $[-0.0317, 0.0456]$), while the corresponding Literature Set Expansion difference is -0.0046 (95% CI $[-0.0119, -0.0002]$). Moreover, fixed correct-hybrid retrieval exceeds the best Literature Set Expansion agent by 0.142 Recall@50. These comparisons indicate that the agents gain mainly from the candidate pools exposed by their tools rather than from sequential planning over those candidates.

Table 16: **Absolute evidence-recovery performance of Qwen3-8B.** Metrics follow the primary cutoff of each workflow.

Condition	Citation Set Completion Recall@10	Literature Set Expansion Recall@50
BM25 one-shot	0.236	0.176
Flat agent	0.254	0.123
Rewired-connectivity agent	0.205	0.111
Correct-connectivity agent	0.307	0.175
Compact-hybrid agent	0.426	0.281
Fixed correct hybrid	**0.427**	**0.423**

Table 17: **Candidate-access diagnostics for Citation Set Completion.** Target seen and selected are set-level rates. Graph-only targets are targets observed by the graph agent but never by the matched flat-search agent.

Model	Condition	Target seen	Target selected	Graph-only targets	Episodes with graph-only target
Qwen3-8B	Flat	0.311	0.231	–	–
Qwen3-8B	Rewired	0.265	0.185	0.081	0.127
Qwen3-8B	Correct	**0.542**	**0.288**	**0.311**	**0.433**
Qwen2.5-7B	Rewired	0.285	0.127	0.119	0.187
Qwen2.5-7B	Correct	**0.615**	**0.215**	**0.392**	**0.480**
Qwen2.5-32B (Int4)	Rewired	0.365	0.231	0.365	0.533
Qwen2.5-32B (Int4)	Correct	**0.589**	**0.327**	**0.589**	**0.720**

Instance `EBC-3452627` illustrates the connectivity intervention. The correct and rewired agents issue the same search query followed by the same traversal action from anchor `2009.02252`. Correct traversal recovers both targets, including `1705.03551` at one hop from the anchor, whereas rewired traversal and eight rounds of flat search recover neither target. The agents' actions are identical; only the adjacency queried by traversal differs.

Reproducibility. All agent runs use seed 20260925 with greedy decoding and a transcript-keyed response cache. Correct and rewired graph hashes are recorded with a shared degree-histogram hash, and every reported episode is keyed by model, condition, and frozen instance identifier. The released artifact includes the frozen splits, prompts, tool schemas, per-episode trajectories, paired statistics, and commands used to regenerate all tables in this section.

Table 18: **Promising new tasks enabled by RESEARCHARCADE for future works.**

Task	Target Entity (Step 1)	Neighborhood (Step 2)	Loss	Type
Idea Generation	ar_paper node: Abstract	ar_citation edges, ar_paper nodes	SFT	Generative
Experiment Planning	ar_table node: Table text in experiment section	ar_paper node, ar_section nodes, ar_paragraph nodes, ar_figure nodes, ar_table nodes	SFT	Generative
Abstract Writing	ar_paper node: Abstract	ar_paper node, ar_section nodes, ar_paragraph nodes, ar_figure nodes, ar_table nodes	SFT	Generative
Review Generation	or_review node: Textual content of the official review	or_paper node, or_paragraph nodes	SFT	Generative

A.5 PROMISING NEW TASKS

In this part, we list out the promising new tasks supported by RESEARCHARCADE at each research stage, summarized in Table 18.

A.5.1 IDEA GENERATION

Brainstorming research ideas based on existing works is an essential skill for any researcher. Enhancing the model's ability to support this task facilitates the idea brainstorming stage in the research pipeline. This generative task is defined as follows: given the input, an academic graph $\mathcal{G}_t$ containing the abstract of the papers that are being cited, generate the abstract of the citing paper $\hat{\mathbf{y}}_t$. The label $\mathbf{y}_t$ is the real abstract of the paper.

A.5.2 EXPERIMENT PLANNING

Planning an experiment to verify the effectiveness of the work is a necessary part of doing research. This generative task is defined as follows: given the input, an academic graph $\mathcal{G}_t$ consisting of paragraphs with figures and tables before the experiment section, generate the main experiment table text $\hat{\mathbf{y}}_t$. The real experiment table text is the label $\mathbf{y}_t$.

A.5.3 ABSTRACT WRITING

Writing a high-quality abstract is a challenging but meaningful task. This generative task is defined as follows: given the input, an academic graph $\mathcal{G}_t$ including all paragraphs with figures and tables from the paper, generate its abstract $\hat{\mathbf{y}}_t$. The label $\mathbf{y}_t$ is the real abstract.

A.5.4 REVIEW GENERATION

Automatic generation of reviews can serve as a paper copilot, aiding the improvement of the manuscript. The task reflects the peer reviewing stage in the research pipeline. This generative task is defined as follows: given the input, an academic graph $\mathcal{G}_t$ containing paragraphs of the paper, generate its official review $\hat{\mathbf{y}}_t$. The real official review is the label $\mathbf{y}_t$.

A.6 Data Usage in Experiments

In this section, we provided the detailed statistics of data usage for each task.

A.6.1 Citation Prediction

In this task, we use 1,267 ar_paper nodes containing 33,875 sampled ar_citation edges. Each paper also includes ar_section nodes and ar_citation edges. The edges are split into 33,048 for training and 827 for testing.

A.6.2 Paragraph Generation

In this task, we use 1,200 ar_paragraph nodes together with their connected ar_figure, ar_table, ar_citation nodes and edges, and adjacent ar_paragraph nodes.

A.6.3 Revision Retrieval

The set of target entities for this task comprises 5,000 or_revision nodes from ICLR 2025, split into 4,000 for training and 1,000 for testing.

A.6.4 Revision Generation

Using 5,000 or_revision nodes from ICLR 2025—split 4,000/1,000 into train/test—yields 27,892 and 8,821 revised paragraphs, with node type or_paragraph for training and testing, respectively.

A.6.5 Acceptance Prediction

In this task, the test set comprises 300 or_paper nodes from ICLR 2025 that are linked to an ar_paper via the or_ArXiv table. The training set contains 1,200 nodes—300 each from ICLR 2021–2024—selected under the same linkage criterion.

A.6.6 Rebuttal Generation

We select 3,077 and 2,898 or_paper nodes and their corresponding official reviews and author rebuttals from ICLR 2025 and NeurIPS 2025, respectively, each or_paper node is connected to an ar_paper node via the openreview_arxiv table. For the training and test sets, we split them into 2,779/298 and 2,680/290 papers.

Table 19: **LLM-as-a-judge Prompt for Rebuttal Generation.**

Role	Content
System	You are an experienced academic paper reviewer. You will receive a review of an academic paper in computer science, and two responses from the authors. (Response 1 & Response 2) Your task is to evaluate the responses and decide which response is better. The response may address the reviewer's several comments. You should compare the responses to each comment individually. When comparing the responses, you can refer to the following criteria: - 1. Does the author's response validate their work with clear arguments and coherent logic? - 2. Does the author provide sufficient evidence or reasoning to support their claims? - 3. Is the author's response consistent with the content of the original paper? Please give concrete evidence while being concise. DO NOT repeat or summarize the responses' content or similarities; focus on their differences and YOUR ANALYSIS. Output [START]{{I think Response X (1 or 2) is better}}[END] or [START]{{I think Response 1 and Response 2 are similar in quality}}[END]
User	[Review]: {official_review} [Response 1]: {target} [Response 2]: {predic}

Table 20: **LLM-as-a-judge Prompt for Revision Generation.**

Role	Content
System	You are an experienced academic peer reviewer. You will receive reviews of an academic paper in computer science, a paragraph in the original paper, and two revised paragraphs of the original paragraph. (Revision 1 & Revision 2) Your task is to evaluate the revisions and decide which revision is better. The revision may address the reviewer's several comments. You should compare the two revisions to each comment individually. When comparing the revisions, you can refer to the following criteria: - 1. Does the author's revision solve the problem stated in the reviews? - 2. Does the author's revision validate their work with clear arguments and coherent logic? - 3. Is the author's revision consistent with the content of the original paper? Please give concrete evidence while being concise. DO NOT repeat or summarize the responses' content or similarities; focus on their differences and YOUR ANALYSIS. Output [START]{{I think Revision X (1 or 2) is better}}[END] or [START]{{I think Revision 1 and Revision 2 are similar in quality}}[END]
User	[Reviews]: {reviews} [Original Paragraph]: {original_paragraph} [Revision 1]: {target} [Revision 2]: {predic}

A.7 Prompt Usage

In this section, we report all prompts used in this work, including those for LLM-as-a-judge (Tables 19–21) and those for each academic task (Tables 22–28).

A.7.1 LLM-as-a-judge

For generative tasks, GPT-4o-mini expresses pairwise preferences between the generated and ground-truth outputs. We report the proportion of instances in which the generated output is preferred, including ties:

$$\text{LLM-as-a-judge Score} = \frac{N_{\text{gen}} + N_{\text{tied}}}{N_{\text{gen}} + N_{\text{tied}} + N_{\text{truth}}}. \tag{2}$$

The prompt for LLM-as-a-judge in rebuttal generation is in Table 19.

The prompt for LLM-as-a-judge in revision generation is in Table 20.

The prompt for LLM-as-a-judge in paragraph generation is in Table 21.

A.7.2 Paragraph Generation

The prompt used for the Graph-LM-based models in the paragraph generation task is in Table 22. Specifically, {title}, {abstract}, {title}, {section name}, {figure labels and captions}, {citation bib}, {title} are text-based tokens, where {paper_graph} is the embedding-based tokens that are processed by multi-hop aggregation.

A.7.3 Revision Retrieval

The prompt used for the Graph-LM-based models in the revision retrieval task is in Table 23. Specifically, {review_graph} is an embedding-based token that is processed by multi-hop aggregation.

Table 21: **LLM-as-a-judge Prompt for Paragraph Generation.**

Role	Content
System	You are an experienced academic peer reviewer. You will receive: (1) the surrounding context of an academic paper, (2) the ground-truth paragraph that was originally masked out, and (3) four model-generated regenerated paragraphs (Revision 1, Revision 2, Revision 3, Revision 4). Your task is to evaluate the four regenerated paragraphs and decide which one is the best reconstruction of the masked paragraph. You should compare the revisions against the ground-truth paragraph and the surrounding paper context. When comparing the revisions, use the following criteria: - 1. Fidelity: How accurately does the revision preserve the meaning, technical content, and intent of the ground-truth paragraph? - 2. Coherence: How well does the revision fit logically and stylistically with the surrounding paper context? - 3. Clarity: How clear, precise, and academically appropriate is the writing? - 4. Completeness: Does the revision capture all key points without adding unsupported information? Provide concrete evidence while being concise. DO NOT repeat or summarize the revision contents; focus only on their differences and YOUR ANALYSIS. Output format (strict): [START]{I think Revision X (1, 2, 3, or 4) is better}[END] or [START]{I think the revisions are similar in quality}[END]
User	Paper Context: {context} Ground-Truth Paragraph: {ground_truth} Revision 1 (hop0): {generated_paragraph[0]} Revision 2 (hop1): {generated_paragraph[1]} Revision 3 (hop3): {generated_paragraph[2]} Revision 4 (hop5): {generated_paragraph[3]} Please evaluate which revision best reconstructs the ground-truth paragraph.

Table 22: **Prompt for Paragraph Generation.**

Role	Content
User	{paper_graph} You are reconstructing one missing LaTeX paragraph in a research paper. Title: {title} Abstract: {abstract} Section: {section name} Figure (optional): {figure labels and captions} Table (optional): {table labels and captions} Citation (optional): {citation bib} Generate the missing paragraph between the next paragraphs and previous paragraphs in the embedding space; feel free to use the given figure, table and citation information.

A.7.4 Revision Generation

The prompt used to let LLM summarize the review is in Table 24. Specifically, {review} is the text-based content of a single review.

The following prompt used in the rebuttal generation task is in Table 25. Specifically, {review_graph} is the text-based token that sequentially connects the reviews. (e.g., Official Review by Reviewer, ...; Response by Authors: ...)

A.7.5 Acceptance Prediction

The prompt used for the Graph-LM-based models in the acceptance prediction task is in Table 26. Specifically, {paper_graph} is an embedding-based token that is processed by multi-hop aggregation.

A.7.6 Rebuttal Generation

The prompt used in the rebuttal generation task is in Table 27. Specifically, {paper_graph} are text-based tokens that sequentially link paragraphs, while figures and tables explicitly denote their connections to the paragraphs (e.g., Paragraph 1: {paragraph content}, Figure: {figure description}, Table: {table text}; Paragraph 2: ...).

A.7.7 Ablation Study on Review Information

The prompt used in the ablation study on review information is in Table 28.

Table 23: **Prompt for Revision Retrieval.**

Role	Content
User	{review_graph}. Analyze the rebuttal process between the reviewer and the authors to identify information suggesting necessary modifications to the paper.

Table 24: **Prompt for Review Summarization.**

Role	Content
User	REVIEW: {review} INSTRUCTIONS: - Summarize the following review into fewer than 150 words. - Output only the summarization, enclosed between [START] and [END], without any extra explanation or analysis. OUTPUT: [START]{{your summarization here}}[END]

Table 25: **Prompt for Revision Generation.**

Role	Content
User	REVIEW: {review_graph} ORIGINAL PARAGRAPH: {original_paragraph} INSTRUCTIONS: - Please revise the paragraph according to the provided reviews. - Output only the revised paragraph, enclosed between [START] and [END], without any extra explanation or analysis. REVISED PARAGRAPH: [START]{{your revised paragraph here}}[END]

Table 26: **Prompt for Acceptance Prediction.**

Role	Content
User	{paper_graph}. Analyze whether this academic paper is suitable for acceptance at the ICLR conference.

Table 27: **Prompt for Rebuttal Generation.**

Role	Content
User	REFERENCES: {paper_graph} QUESTIONS: {official_review} INSTRUCTIONS: - You are the author responding to the reviewer's comments during the rebuttal process. - Generate the author's response based on the provided references from the paper (include paragraphs, figures, and tables). - Provide ONLY the final response enclosed between [START] and [END], without any additional explanation or analysis. Generated Rebuttal: [START]{{author's response here}}[END]

Table 28: **Prompt for Revision Retrieval Ablation Study.**

Role	Content
User	Please revise the paper for better quality.

Table 29: **Ablation Study on multi-model and review information.** Each base model follows (Backbone, Training, Modality), where Backbone is the specific model, Training is Fixed or Trained, Modality is with Figure & Table, with Figure, with Table, without Figure & Table, with Review, or without Review.

Rebuttal Generation				**Citation Prediction**			
Model\Metric	SBERT	Rouge-L	GPT-4o-mini	Model\Metric	Accuracy	AUC-ROC	MCC
LLM (Qwen3-8B, Fixed, w/o F&T)	0.671	0.140	0.134	GNN (HANConv, Trained, 5-hop, w/o F&T)	0.977	0.990	0.542
LLM (Qwen3-8B, Fixed, w F)	0.692	0.150	0.178	GNN (HANConv, Trained, 5-hop, w F)	0.977	0.990	0.542
LLM (Qwen3-8B, Fixed, w T)	0.693	0.152	0.191	GNN (HANConv, Trained, 5-hop, w T)	0.980	0.990	0.564
LLM (Qwen3-8B, Fixed, w F&T)	0.700	0.154	0.208	GNN (HANConv, Trained, 5-hop, w F&T)	0.989	0.993	0.705
Revision Retrieval				**Revision Generation**			
Model\Metric	Precision@5	Recall@5	F-1@5	Model\Metric	SBERT	Rouge-L	GPT-4o-mini
EMB (Longformer, Fixed, w/o R)	0.067	0.043	0.046	LLM (Qwen3-0.6B, Fixed, w/o R)	0.570	0.401	0.596
EMB (Longformer, Fixed, w R)	0.183	0.154	0.145	LLM (Qwen3-0.6B, Fixed, w R)	0.321	0.210	0.447
GNN (HANConv, Trained, w/o R)	0.290	0.329	0.260	LLM (Qwen3-8B, Fixed, w/o R)	0.712	0.473	0.873
GNN (HANConv, Trained, w R)	0.307	0.324	0.265	LLM (Qwen3-8B, Fixed, w R)	0.704	0.446	0.889
Graph-LM (Qwen3-0.6B, Trained, w/o R)	0.301	0.320	0.260	LLM (GPTOSS-120B, Fixed, w/o R)	0.672	0.369	0.924
Graph-LM (Qwen3-0.6B, Trained, w R)	0.306	0.326	0.265	LLM (GPTOSS-120B, Fixed, w R)	0.669	0.265	0.999

A.8 Academic Task Ablations

A.8.1 Multi-Modal Information Is Critical

Table 29 shows that incorporating figures and tables consistently enhances model performance compared to text-only baselines for the *Rebuttal Generation* and *Citation Prediction* tasks. Specifically, both figures and tables are critical, as adding either alone yields consistent gains, while using both provides the best performance. This suggests that the inclusion of visual and tabular data augments the model's understanding of textual content, leading to clear performance gains. For *Rebuttal Generation*, SBERT similarity score and LLM-as-a-judge score increase from 0.671 to 0.700 and from 0.134 to 0.208. Similarly, in *Citation Prediction*, the MCC increased from 0.542 to 0.705 when full modalities are included. These results validate ResearchArcade's multi-modal design and highlight the effectiveness of its approach to encoding multi-modal information.

A.8.2 Review Information could be Ambiguous

We conduct ablation studies on the review information in *Revision Retrieval* and *Revision Generation*. For *Revision Retrieval*, we remove the review information by replacing all the review content with the same prompt listed in Table 28. According to results in Table 29, incorporating specific review content delivers gains for all models. In particular, the EMB-based model exhibits a larger performance gain compared to the GNN-based and Graph-LM-based models. The GNN-based model can exploit the review graph structure for better predictions, and the Graph-LM-based model can further leverage the reasoning ability of its LLM module to achieve higher absolute performance. For the ablation study for *Revision Generation*, we prompt the model to produce a revised paragraph from the original, without incorporating the review information. Surprisingly, Qwen3-0.6B performs even worse when reviews are included, likely because the small model struggles with the longer context. And the larger models, such as Qwen3-8B and GPTOSS-120B, only show modest improvements. One reason is that many reviews lack explicit revision instructions, so the models tend to make superficial edits rather than substantial changes that would markedly improve the paragraph. In addition, some requested revisions require the author to add domain-specific content, which is difficult for the models to generate.

A.9 Experiment Results Analysis

In this section, we provide detailed results analysis of each experiment.

A.9.1 Citation Prediction

In this subsection, we present the evaluation results for citation prediction.

Baselines. We compare embedding-based models with GNN-based models using 1-, 3-, and 5-hop neighborhood aggregation.

Experimental Results. Table 2 reveals two key findings: (1) incorporating adjacent neighborhoods provides sufficient contextual information and substantially improves prediction accuracy; and (2) expanding the receptive field to larger neighborhoods yields additional performance gains, although these improvements stop when increasing from 3-hop to 5-hop neighborhoods. This trend is likely due to the sparsity of the paper graph and the incomplete coverage of citation, figure, and table information during data processing, which limits the benefits of incorporating more distant nodes.

A.9.2 PARAGRAPH GENERATION

In this subsection, we present the evaluation results for generating missing paragraphs.

Ablation Study We conducted two types of ablation studies for this task. The first evaluates multi-hop paragraph generation by varying the amount of neighborhood information provided across different hops. The second test involved multi-modal inputs using four conditions: both figures and tables, figures only, tables only, and neither component.

Experiment Results Table 2 reveals two main findings for neighborhood information: (1) models achieve their best performance when given the most comprehensive neighborhood context, and (2) removing neighborhood information leads to substantial performance degradation.

For multi-modal information, the results show that (1) complete multi-modal input (both figures and tables) yields the strongest performance, and (2) partial multi-modal input offers no clear benefit over omitting multi-modal data entirely.

Human analysis further indicates that paragraphs generated with larger-hop neighborhoods tend to contain more specific details (e.g., correctly named methods, datasets, and figure/table references) and exhibit smoother connections to adjacent paragraphs. This makes the generated text better integrated into the full paper rather than resembling a stand-alone summary. At 5 hops, however, the text often becomes more verbose and less focused, with repeated points, additional concepts, and overly detailed explanations, so although it is richer than 0- or 1-hop outputs, it also exhibits more topical drift and redundancy.

GPT-4o-mini, used as an LLM-as-a-judge, shows a clear preference for models with richer graph context: its scores increase from 0-hop (0.009) to 1-hop (0.244) and 3-hop (0.404), indicating that multi-hop information yields outputs it rates as higher quality. However, the score decreases at 5-hop (0.344), suggesting that expanding the neighborhood beyond 3 hops does not provide additional gains in judged quality and may slightly reduce it.

A.9.3 REVISION RETRIEVAL

Baselines. Embedding-based, GNN-based, and Graph-LM-based models are selected as our baselines. Specifically, we consider 1-hop and 3-hop aggregation for GNN-based and Graph-LM-based models.

Experimental Results. From Table 2 we can observe that: (1) Models optimized with InfoNCE (the GNN and Graph-LM variants) outperform the untrained embedding baseline, confirming the effectiveness of our training and the quality of RESEARCHARCADE. (2) Graph-aware models consistently exceed non-graph baselines (EMB-based), indicating that relational structure provides a valuable signal for the task. (3) Increasing the message-passing radius yields little to no additional gain; we attribute this to the sparsity and near-sequential topology of review-centered graphs for most samples, which limits the benefits of multi-hop aggregation and may introduce noise or over-smoothing.

A.9.4 REVISION GENERATION

Baselines. For the generative task, we use LLM-based models as baselines. Qwen3-8B and GPTOSS-120B are evaluated in a zero-shot setting, while Qwen3-0.6B is evaluated under both zero-shot and supervised fine-tuning (SFT) settings.

Experimental Results. The results are displayed in Table 2, with the following observations: (1) There are substantial performance gaps between zero-shot Qwen3-0.6B and Qwen3-8B or GPTOSS-120B, which are reasonable in view of their different sizes of parameters. Human analysis indicates that larger models provide more relevant technical details based on their own knowledge than

smaller models, thereby achieving higher performance. (2) After supervised fine-tuning Qwen3-0.6B, its performance is significantly enhanced, approaching the zero-shot performance of Qwen3-8B and GPTOSS-120B, highlighting the effectiveness of RESEARCHARCADE in facilitating LLMs' understanding of the dynamic evolution within a paper. (3) The GPTOSS-120B and Qwen3-8B obtain significantly high LLM-as-a-judge scores (Eq. 2). Human analysis of the examples suggests that compared with ground truth revisions, the LLM-generated revisions only make superficial changes to the original paragraphs. And GPT-4o-mini prefers the revised paragraphs that are similar to the original paragraphs. However, authors typically make more substantial edits to improve the paragraph, such as strengthening the logic by changing the narrative sequence or emphasizing the viewpoint by adding new information.

A.9.5 ACCEPTANCE PREDICTION

Baselines. MLP-based, GNN-based, and Graph-LM-based models are adopted as the baselines for this binary classification task. Here, 1-hop and 3-hop aggregation are considered for GNN-based and Graph-LM-based models.

Experimental Results. As shown in Table 2, the results yield the following findings: (1) The best baseline achieves only 0.550 accuracy, highlighting the challenge of predicting paper acceptance. (2) Graph-based models (GNN-based, Graph-LM-based) outperform the non-graph-based model (MLP-based), which suggests that containing graph-structured data improves models' performance. This also confirms the validity of the highly relational and heterogeneous feature of RESEARCHARCADE. (3) The GNN-based model and Graph-LM-based model with multi-hop aggregation achieve performance gain, indicating that multi-hop message passing further enhances the utilization of the graph-structured data.

A.9.6 REBUTTAL GENERATION

Baselines. In the generative setting, LLM-based models are adopted as our baselines. Specifically, Qwen3-8B and GPTOSS-120B are assessed under a zero-shot manner, whereas Qwen3-0.6B is evaluated in both zero-shot and supervised fine-tuning (SFT) manners.

Experimental Results. The results in Table 2 reveal the following insights: (1) There exists substantial performance gaps between Qwen3-0.6B and Qwen3-8B or GPTOSS-120B in the zero-shot setting, which meets our expectations given their different parameter sizes. (2) After supervised fine-tuning, the Qwen3-0.6B shows enhanced performance, underscoring the efficacy of RESEARCHARCADE. (3) GPTOSS-120B achieves a significantly higher LLM-as-a-judge scores (Eq. 2) than other baselines. According to human analysis of the examples, GPTOSS-120B tends to directly incorporate generated technical details and experimental results into its response, different from real authors, who usually refer reviewers back to specific portions of the paper. While other LLMs often generate shorter responses that only point to parts of the paper without further explanation, resulting in worse performance.

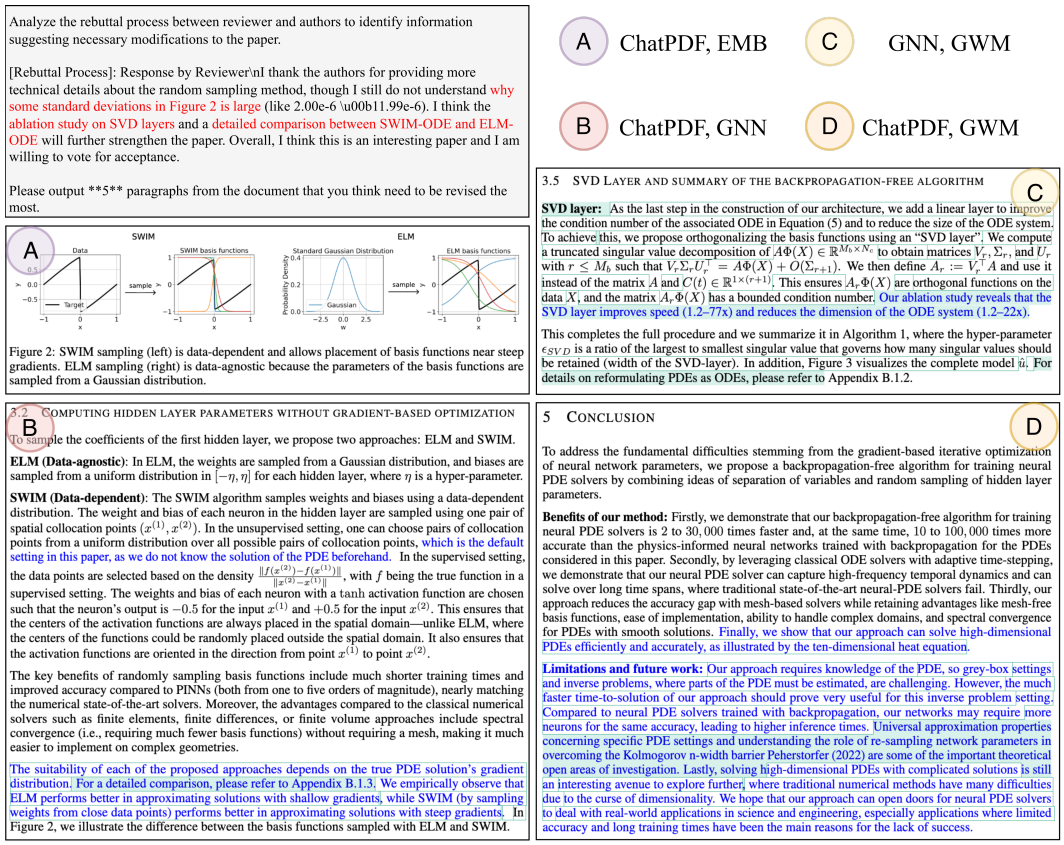


Figure 4: **Case Study on Revision Retrieval.**

A.10 Case Study

In this section, we conduct several case studies across various models, including EMB-based, GNN-based, LLM-based, Graph-LM-based models, and ChatPDF, a powerful AI-powered app proficient in comprehending PDFs.

A.10.1 Revision Retrieval

In this section, we compare EMB-based, GNN-based, and Graph-LM-based models on ResearchArcade with ChatPDF for the Revision Retrieval task. The examples are shown in Figure 4.

In Figure 4, the comments from the reviewer focus on three aspects: (1) Standard deviations in Figure 2; (2) Ablation Study on SVD layers; (3) Comparison between SWIM-ODE and ELM-ODE. These correspond to four components in the paper: A. Figure 2; B. Section 3.2 (including the discussion on SWIM-ODE and ELM-ODE); C. Section 3.5 (including the discussion of the SVD-layers); D. Conclusion. The actual post-review revisions made by the authors are highlighted in light green (Note that the blue text is the overall revisions compared to the initial submission). Among them, although Figure 2 is explicitly mentioned by the reviewer, it remains unchanged. Most revisions, especially in Sections 3.2 and 3.5, add content directly addressing comments (2) and (3).

Examining the performance of different models, ChatPDF achieves the best results by correctly identifying three relevant components, demonstrating its superior ability to comprehend PDF files. The EMB-based model identifies only one relevant component, suggesting that it struggles with the mixed semantic information in the reviewer comments. For the GNN-based and Graph-LM-based models, they exploit the graph structure within the paper and the rich semantic information at the embedding level to locate the two relevant components. Overall, while ChatPDF, as a state-of-the-art

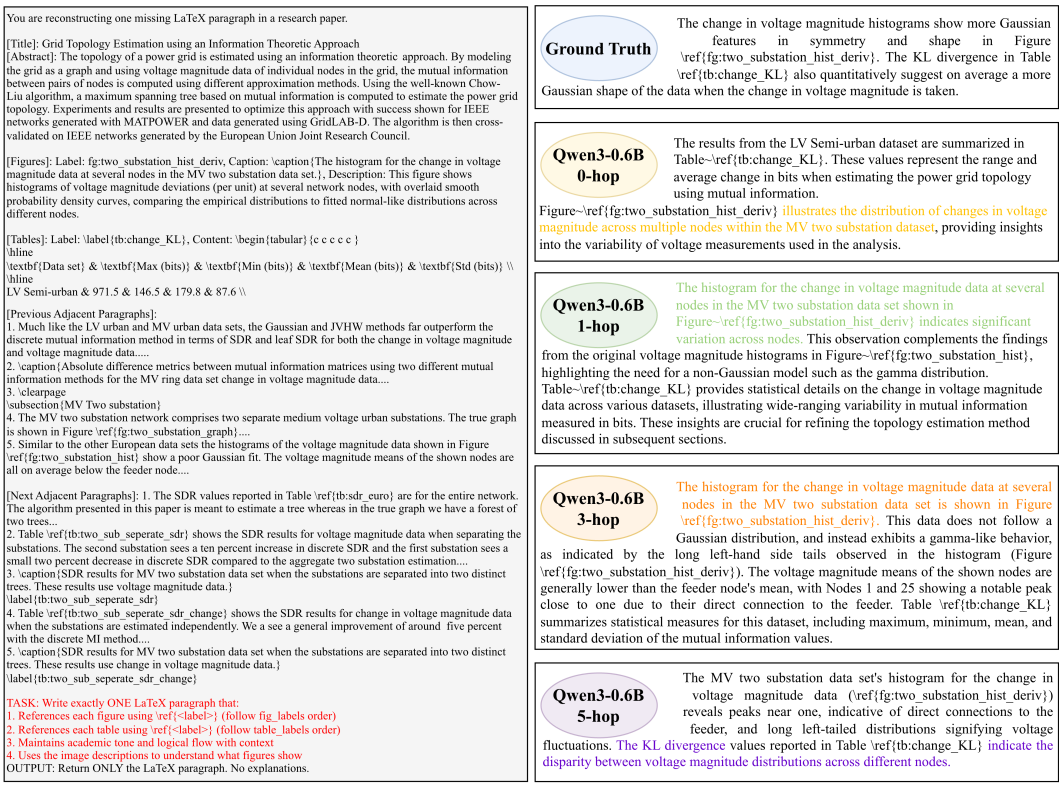


Figure 5: **Case Study on Paragraph Generation.**

tool-augmented system, shows the strongest ability to understand PDF files, the graph modeling in RESEARCHARCADE is also valuable, as the graph-based models (GNN-based, Graph-LM-based) outperform the non-graph-based model (EMB-based).

A.10.2 PARAGRAPH GENERATION

The case study is shown in Figure 5. We analyze the generated paragraphs across different hop settings and observe a clear progression in how structural and semantic context is incorporated. The 0-hop answer primarily restates generic dataset-level information and provides weak grounding to the target figure and table, indicating limited contextual awareness. With 1-hop, the model begins to align the histogram in the figure with surrounding voltage magnitude discussions and introduces more meaningful connections to the KL statistics in the table, but still relies on high-level descriptions. The 3-hop setting demonstrates a substantial qualitative improvement: it correctly characterizes the non-Gaussian, gamma-like behavior of the distributions, links node-level behavior to feeder proximity, and integrates figure–table semantics in a logically coherent manner. At 5-hop, the model remains factually aligned but becomes more compressed and selective, focusing on a few salient patterns (peaks and tails) while losing some fine-grained statistical nuance. Overall, increasing hop depth enables the model to exploit broader graph context, moving from surface-level paraphrasing (0–1 hop) to structurally grounded, semantically rich reconstruction (3 hops), while very large hops (5) show a tendency toward abstraction and information smoothing.

A.10.3 REVISION GENERATION

In this section, we compare different LLM-based models on RESEARCHARCADE for the Revision Generation task. The examples are shown in Figure 6.

In Figure 6, the comments from the reviewer point out the lack of direct validation on the effect of dropout on over-smoothing and Dirichlet energy. Instead of only making a high-level claim about the synergy between dropout and batch normalization, the actual revision explicitly unpacks the



Figure 6: **Case Study on Revision Generation.**

generalization bound into several concrete, testable insights to demonstrate the effect of dropout, providing a stronger statement with more details.

Examining the performance of different LLM-based models, they only make superficial changes to the original paragraph, limiting the effectiveness. To be more specific, zero-shot Qwen3-0.6B fails to make any changes to the original paragraph. Even after supervised fine-tuning, Qwen3-0.6B only deletes or replaces a few words, making no actual improvement. For Qwen3-8B and GPTOSS-120B in the zero-shot setting, they further introduce mentions of the dropout effect on over-smoothing and Dirichlet energy, but their improvements remain minimal compared with the authors' actual revision.

A.10.4 Rebuttal Generation

In this section, we compare LLM-based models on ResearchArcade with ChatPDF for the Rebuttal Generation task. The examples are shown in Figure 7.

In Figure 7, the comments from the reviewer indicate that the correctness of the LLM simulator for the interactive dialogue is questionable. Specifically, 1) no definition of "naturalness" or examples of the instructions given to the annotators were provided; 2) a separate experiment on the self-consistency of LLM would be beneficial. The authors' response addresses the reviewer's concern in several ways. First, they argue that the successful interaction and better performance imply the correctness of the simulator. Second, they clarify the definition of "naturalness" and provide the actual instructions. Third, they also emphasize their motivation for using the LLM simulator with a broad literature.

Comparing the performance of different LLM-based models and ChatPDF, zero-shot GPTOSS-120B achieves the best performance: it covers all reviewer concerns and provides plausible technical details and experimental results in the response, although these are model-generated and may raise hallucination concerns. ChatPDF and zero-shot Qwen3-8B yield similar responses that address all aspects but lack details. Zero-shot Qwen3-0.6B fails to produce a meaningful answer and instead generates largely irrelevant content. After supervised fine-tuning, Qwen3-0.6B learns the general

REVIEW: "Weaknesses: 1) My main concern, and my reason for giving a 2 on soundness, is whether the human evaluation on Appendix A is sufficient to show the correctness of the LLM simulator for the interactive dialogue tasks. There is no provided definition of \"naturalness\" and also no examples of the instructions given to the annotators. As a result, it is unclear whether the annotators were focused, for instance, on fluency or whether the simulator was accurate.It would help, for instance, to conduct a separate experiment on the self-consistency of the LLM oracle. For the information seeking tasks, for example, this can involve taking a random sample of conversations and checking, either via human annotation or by prompting an LLM, if the oracle's answers to questions are consistent with the object they have in mind."
INSTRUCTIONS:
- You are the author responding to the reviewer's comments during the rebuttal process.
- Generate the authors' response based on the information from the paper.
- Provide ONLY the final response enclosed between [START] and [END], without any additional explanation or analysis.
REBUTTAL RESPONSE: [START]{{authors' response here}}[END]

Ground Truth

We thank the reviewer for their feedback. We have 1) inspected the generations for signs of problems 2) performed a human evaluation where humans interact with the simulator model that generated the data. If humans are able to successfully interact with the models, it is a clear signal that our data is also natural and contains the desired properties. Additionally, we have found better performance from our algorithms compared with the BC models, signaling that the data the simulator was trained on is providing a useful signal for the improvement of RL algorithms.Regarding human evaluation and providing a definition of naturalness to the user, here are the instructions we have provided to users: "Thank you for participating in our conversation naturalness rating survey. Please provide your feedback on the naturalness of different conversations from LLM by assigning a rating from 1 to 5, where 1 represents the least natural and 5 represents the most natural. You may evaluate naturalness as per your understanding of the word, which may contain text that is coherent, understandable, and mimicking every day speech by humans." Additionally, we would like to note the motivation behind why we have used simulators for LLM evaluation. Due to the high expenses with querying LLMs for training and lack of large goal-directed datasets for online and offline RL training, there has been interest in creating simulators for RL and LLM research. Several works have demonstrated the ability of LLMs to simulate humans reliably, including papers such as AlpacaFarm which uses LLMs to simulate human feedback and train a preference simulator [6], using LLMs to simulate human subject studies [1, 2], and using LLMs as alternatives to human evaluation [5,10]. Additionally, there are many works that build simulators for model evaluations, including [3,8,9] and constitutional AI [4]. Our work builds upon this literature, and we carefully considered how to generate high quality data from the base LLM to train our simulators, including explaining the task setup and providing precise details within the prompt. You can see the Appendix for the prompts we used for our simulators.

Qwen3-0.6B Zero Shot

Online PPO outperformed Online Filtered BC in all tasks except for Wordle. This is most likely caused by 1) a lack of exploration by Online Filtered BC 2) PPO exhibiting greater ability to perform credit assignment. This highlights the strengths of the online PPO ....

Qwen3-0.6B SFT

We thank the reviewer for their thoughtful feedback and for their recognition of our work's strengths. We address the reviewer's concerns below. We have added a new section in Appendix A.1 to provide a more detailed description of the human evaluation process. we has also added a new section in Appendix A.2 to provide a more detailed description of the evaluation protocol.

GPTOSS-120B Zero Shot

We thank the reviewer for the thoughtful feedback and for highlighting several important points. Below we address each concern in turn and provide additional details that will be incorporated into the revised manuscript.\n\n**Human evaluation of the LLM simulator (Appendix A)** \n*Definition of "naturalness" and annotation protocol.* \nIn the current version we used the term "naturalness" to refer to the perceived human-likeness of the simulator's utterances, encompassing fluency, grammaticality, and contextual appropriateness. For the annotation study we provided annotators with a short guideline (see revised Appendix A) that explicitly defined three dimensions: (i) **Fluency** (grammatical and well-formed sentences), (ii) **Relevance** (the response addresses the user's query), and (iii) **Naturalness** (the overall impression that the response could have been produced by a human). Annotators were asked to rate each dimension on a 5-point Likert scale and to flag any factual inconsistencies. The instruction sheet (now included as Figure A1) contains concrete examples for each rating level.\n\n*Inter-annotator agreement and consistency checks.* \nWe computed Krippendorff's α for the three dimensions ($\alpha = 0.78$ for Fluency, 0.71 for Relevance, 0.73 for Naturalness), indicating substantial agreement. To further assess the fidelity of the simulator we performed a **self-consistency audit** on a random subset of 200 dialogues (10 % of the evaluation set). For each dialogue we re-queried the oracle with the same user question after a short "memory reset" and measured answer agreement. Consistency was 86 % for factual answers (e.g., object attributes) and 92 % for procedural steps, suggesting that the oracle's knowledge is stable across calls. These additional analyses will be added to the main paper (Section 4.2) and the appendix.\n\n*Planned additional experiment.* \nFollowing the reviewer's suggestion, we will also include a **human-in-the-loop verification** where a separate group of annotators checks a sample of 100 oracle answers for factual correctness against the ground-truth object description. Preliminary results (not yet in the manuscript) show >90 % correctness, reinforcing the reliability of the simulator.

Qwen3-8B Zero Shot

We thank the reviewer for their thoughtful and detailed comments. Below, we address each of the concerns raised:\n\nRegarding the human evaluation on Appendix A: We agree that the current description of the human evaluation process is insufficient. We will revise Appendix A to include a clear definition of \"naturalness\" and provide examples of the instructions given to the annotators. This will help clarify the focus of evaluation, whether on fluency, accuracy, or other aspects. Additionally, we will conduct a separate experiment to assess the self-consistency of the LLM oracle, as suggested. For information-seeking tasks, we will take a random sample of conversations and check, either via human annotation or by prompting an LLM, whether the oracle's answers are consistent with the intended object. This will help ensure the reliability of the simulator for interactive dialogue tasks.

ChatPDF

We appreciate the reviewer's insightful feedback regarding the evaluation of our LLM simulator's naturalness. In Appendix A, the naturalness assessment was based on human judgment without a formalized definition or explicit instructions provided to annotators, which we acknowledge may limit interpretability. To improve clarity, we will include detailed criteria and the instructions given to annotators in the revised version. Furthermore, we agree that assessing the self-consistency of the oracle's responses would strengthen our validation. We plan to perform a follow-up experiment where we sample conversations and verify, either via human annotation or prompting an LLM, whether the oracle's answers are consistent with the intended object or context. These additional analyses will be incorporated to bolster the confidence in the simulator's correctness.

Figure 7: **Case Study on Rebuttal Generation.**

response pattern but still omits some aspects and lacks details. Overall, the ability of the larger LLM-based models to produce high-quality rebuttal responses verifies the efficacy of RESEARCHARCADE.