RESEARCHARCADE: GRAPH INTERFACE FOR ACADEMIC TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Academic research generates diverse data sources. As researchers increasingly use machine learning to assist research tasks, a crucial question arises: *Can we build a unified data interface to support the development of machine learning models for various academic tasks?* Models trained on such a unified interface can better support human researchers throughout the research process and eventually accelerate knowledge discovery. In this work, we introduce RESEARCHARCADE, a graph-based interface that connects multiple academic *data sources*, unifies *task definitions*, and supports a wide range of *base models* to address key academic challenges. RESEARCHARCADE utilizes a coherent multi-table format with graph structures to organize data from different sources, including academic corpora from ArXiv and peer reviews from OpenReview, while capturing information with multiple modalities, such as text, figures, and tables. RESEARCHARCADE also preserves temporal evolution at both the manuscript and community levels, supporting the study of paper revisions as well as broader research trends over time. Additionally, RESEARCHARCADE unifies diverse academic task definitions and supports various models with distinct input requirements. Our experiments across six academic tasks demonstrate that combining cross-source and multi-modal information enables a broader range of tasks, while incorporating graph structures consistently improves performance over baseline methods. This highlights the effectiveness of RESEARCHARCADE and its potential to advance research progress.

1 INTRODUCTION

Academic research represents a pinnacle of human knowledge discovery. Diverse research tasks such as forecasting research trends and debugging scientific papers (Sundar et al., 2024; Tian et al., 2025; Feng et al., 2025a) demand access to comprehensive data from multiple sources. To accomplish these tasks, various models are employed. These complexities raise an important research question: *Can we build a unified data interface to support the development of machine learning models for various academic tasks?*

Building such an interface for research tasks is challenging. In terms of data, firstly, academic data is sourced from diverse platforms such as ArXiv and OpenReview, encompassing complex relationships among entities like authors, papers, citations, and reviews. This requires a flexible framework capable of managing highly relational data. Secondly, the data representations themselves span multiple modalities—from textual content to visual and tabular data. Holistically integrating these varied representations is a significant challenge. Additionally, the dynamic and ever-evolving nature of academic data further complicates the task, as continuous growth and maintenance of the framework are required to keep pace with ongoing research developments. In terms of tasks, defining different academic tasks demands significant effort in data preprocessing and task formulation. In terms of models, different types of models require distinct interfaces. For example, Large Language Models (LLMs) require text-based data as input, while Graph Neural Networks (GNNs) utilize graph-structured data.

Despite existing efforts to benchmark scientific research, developing a unified and dynamic representation of research activities remains an open challenge. While existing academic datasets have systematically collected and organized academic data (Kang et al., 2018; Lo et al., 2019), they

mainly focus on single-source data, such as academic corpora or peer reviewing conversations. Although multi-modal data (e.g., figures and tables within scientific papers) have been incorporated to construct valuable datasets (Xia et al., 2024; Tian et al., 2025), these approaches do not fully exploit the multi-modal relations among different data types. Recent works have used graphs to model academic data and define academic tasks (Li & Tajbakhsh, 2023; Zhang et al., 2024). However, each academic task is still formulated individually, requiring repetitive developmental efforts.

In this paper, we propose RESEARCHARCADE, a graph-based interface that links diverse academic *data sources*, with unified *task definitions*, and supports a large variety of *base models* to solve valuable academic tasks. Overall, RESEARCHARCADE exhibits four core features that make it ideal for solving academic tasks: *Multi-Source*, *Multi-Modal*, *Highly Structural and Heterogeneous*, and *Dynamically Evolving*. RESEARCHARCADE integrates academic data from multiple sources, including research papers from ArXiv and peer reviews with revisions from OpenReview, while collecting multi-modal information, including text, figures, and tables. These distinct entities are organized in a coherent multi-table format, with selected tables designated as nodes and edges, enabling RESEARCHARCADE to efficiently handle the highly relational and heterogeneous data as graphs within academic communities. Moreover, RESEARCHARCADE models academic evolution at two scales: microscopically, it preserves paper revisions with temporal information to track individual manuscript development, and macroscopically, its extensible framework enables continuous data incorporation, supporting analysis of research trends over time. Furthermore, we unify diverse academic tasks within the academic graphs in RESEARCHARCADE, enabling straightforward formulation of new tasks across both predictive and generative paradigms. Additionally, the structured knowledge in RESEARCHARCADE can be easily exported to standardized formats, such as CSV and JSON, facilitating integration with various models, including LLMs and GNNs.

To demonstrate the key advantages of RESEARCHARCADE, we define six academic tasks: figure/table insertion, paragraph generation, revision retrieval, revision generation, acceptance prediction, and rebuttal generation. Extensive experiments show that models benefit from the multi-source, multi-modal, heterogeneous, and dynamic information in RESEARCHARCADE.

Overall, our key contributions include: First, RESEARCHARCADE enables diverse task definitions by integrating multiple data sources, multi-modal information, and supporting the inclusion of temporal and up-to-date data. Second, RESEARCHARCADE facilitates the academic task solving by unifying the task formulations and supporting the training of various models. Finally, RESEARCHARCADE shows that incorporating graph structures consistently enhances model's performance compared to baseline approaches.

2 RELATED WORK

Academic data as graphs. Existing research on academic graphs employs various decompositions on academic data. UNARXIVE (Saier et al., 2023) and DOCGENOME (Xia et al., 2024) model academic corpora by representing papers, paragraphs, and citations as nodes, while also extracting tables and figures. OAG-BENCH (Zhang et al., 2024) models academic communities as heterogeneous graphs, defining nodes such as authors, papers, and affiliations. In RESEARCHARCADE, we integrate these entities and extend with heterogeneous graphs.

Dynamic modeling of academic data. Academic evolution is broadly classified into two parts: research trends and individual manuscript evolution. Several existing works focus on analyzing the evolution of research trends. Gollapalli & Li (2015) analyzes twenty years of ACL and EMNLP proceedings using topic distributions to trace venue convergence and divergence, while Tian et al. (2023) models scientific subcommunity evolution as event prediction, detecting growth, splits, and merges in collaboration graphs. These works focus on inter-paper evolution, while intra-paper evolution remains unexplored.

Solving academic tasks with deep learning. Various deep learning models are utilized to solve the academic tasks. Zhang et al. (2024) leveraged CNNs, GNNs, and LLMs to solve diverse academic tasks. However, their efforts are scattered and require highly specialized models. RESEARCHARCADE offers a general graph interface to unify input data and task definitions for academic tasks, providing a platform for addressing various academic challenges.

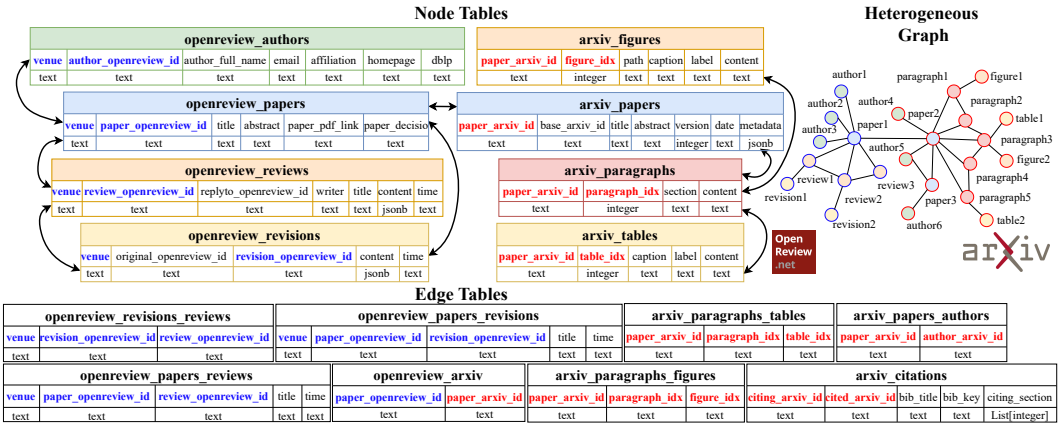

Figure 1: **RESEARCHARCADE uses a multi-table format with graph structures to collect data from different sources with multiple modalities.** Tables are classified into node tables (colored) or edge tables (black and white). The blue (denoting the OpenReview part) or red (denoting the ArXiv part) columns represent the unique identification of each node or edge, and the remaining columns represent the features of the nodes or edges. The conversion from the multiple tables to heterogeneous graphs is straightforward.

3 RESEARCHARCADE DATA DESCRIPTION

RESEARCHARCADE is an inclusive mapping of real-world research knowledge, featuring four key attributes: (1) multi-source, (2) multi-modal, (3) highly relational and heterogeneous, (4) dynamically evolving. An overview is illustrated in Figure 1, with further details in Appendix Figure 3.

3.1 MULTI-SOURCE & MULTI-MODAL

RESEARCHARCADE is primarily sourced from computer science papers in ArXiv and submissions from The International Conference on Learning Representation (ICLR) in OpenReview. Beyond text-based data, RESEARCHARCADE also integrates multi-modal data (e.g., figures and tables), supporting more complex multi-modal tasks.

ArXiv: RESEARCHARCADE includes 40,210 papers from ArXiv across 40 computer science categories, comprising 4,984,166 paragraphs, 503,775 figures, 198,957 tables, and 1,554,055 citations. Relevant connections between these entities are also captured by RESEARCHARCADE. Detailed statistics are provided in Table 5, and the procedure of data collection is in Appendix A.2.1.

OpenReview: RESEARCHARCADE also includes data from OpenReview, which comprises 28,648 submissions from all ICLR conferences throughout history, contributed by 88,351 authors. In addition, the resulting 472,448 reviews and 54,467 submission revisions during the rebuttal process are included. These entities are enriched with valuable connections. Detailed statistics are given in Table 6, and the step-by-step data collection procedure is described in Appendix A.2.2.

Connect ArXiv and OpenReview: Connecting the data from the ArXiv and OpenReview contributes to more comprehensive academic graphs, allowing the definition of more diverse academic tasks. To achieve this goal, each submission in OpenReview is associated with its corresponding paper in ArXiv based on the title. The statistics are shown in Table 7.

3.2 HIGHLY RELATIONAL AND HETEROGENEOUS

Research activities in academic communities are modeled by interactions among typed entities. RESEARCHARCADE stores data in a multi-table node–edge schema, consisting of node tables and edge tables, which directly map to heterogeneous graphs. An illustration is shown in Figure 1.

Using data from ArXiv, RESEARCHARCADE constructs a two-scale graph representation of the literature. At the intra-paper level, each paper is decomposed into a paragraph-scale content graph including paper, paragraphs, figures, and tables nodes, linked by typed edges (e.g., paper-paragraph,

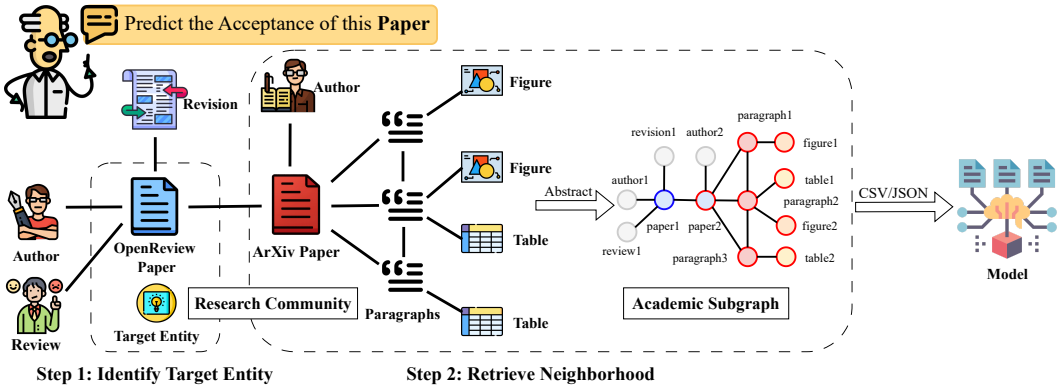

Figure 2: **RESEARCHARCADE unifies the academic task definitions in a two-step scheme**: (i) Label: Identify the task's target entity and assign its attribute as label; (ii) Input: Retrieve the target entity's neighborhood to construct an academic graph that supports task solving.

paragraph-figure/table). At the macro inter-paper level, we include authors, subject categories, and citation links, adding edges for authorship, category assignment, and paper-to-paper citations.

The academic graphs built on data from OpenReview mainly model the academic activities that happen during the peer review process. It encompasses diverse types of nodes, such as papers, authors, paragraphs, reviews, and revisions. Some key relationships are also included: the authorship, which connects papers and authors; the comment-under-paper relation, which connects papers and reviews; the revision-of-paper relation, which connects papers and revisions; the revision-caused-by-review relation, which connects reviews and revisions, etc.

3.3 DYNAMICALLY EVOLVING

As the academic community continuously evolves, RESEARCHARCADE records temporal information (e.g., paper upload dates and paper revision timestamps), enabling a realistic simulation of scholarly dynamics. This includes tracing the evolution of research trends and modeling paper updates driven by the rebuttal process. Moreover, RESEARCHARCADE can be continuously updated to reflect the ongoing development in the academic community.

4 ACADEMIC TASKS ON RESEARCHARCADE

Defining different academic tasks often requires repetitive work, such as data collection, cleaning, and task specification. With RESEARCHARCADE, these tasks can be unified and conveniently defined on our academic graphs.

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 an academic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which is heterogeneous. In this context, 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 ACADEMIC TASK DEFINITION

As is shown in Figure 2, RESEARCHARCADE unifies the academic task definitions in the following two steps: (1) identifying the target entity and (2) retrieving the neighborhood of the target entity.

Table 1: **Summary of six academic tasks studied with RESEARCHARCADE.** Abbreviations include "or": openreview, "ar": ArXiv, CE: Cross-entropy Loss, BCE: Binary Cross-entropy Loss.

Task	Target Entity (Step 1)	Neighborhood (Step 2)	Loss	Type
Figure/Table Insertion	ar_paragraph_figure/table node: Index list of parent paragraphs	ar_section nodes, ar_paragraph nodes, ar_table nodes, ar_figure nodes, ar_citation edges	CE	Predictive
Paragraph Generation	ar_paragraph node: Textual paragraph 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

Step 1: Identifying the target entity of an academic task. The target entity is either a node v or an edge e, with attributes that define the labels for the task. Let t denote the target entity with attributes $\mathbf{a}_t$. Its certain attributes, denoted as $\mathbf{y}_t \subseteq \mathbf{a}_t$, are the labels implied in the task.

Step 2: Retrieving the neighborhood of the target entity. To support the academic task solving, the multi-hop neighborhood of the target entity t is retrieved, constructing an academic graph $\mathcal{G}_t$ centered at t. The one-hop neighborhood $\mathcal{N}_t^{(1)}$ of t consists of entities directly connected to t. If $t \in \mathcal{V}$, then $\mathcal{N}_t^{(1)} = \{k \,|\, k \in \mathcal{V}, (t, k) \in \mathcal{E}\}$. If $t \in \mathcal{E}$, then $\mathcal{N}_t^{(1)} = \{k, u \,|\, k, u \in \mathcal{V}, t = (k, u)\}$. For $i > 1$, the i-hop neighborhood is defined as $\mathcal{N}_t^{(i)} = \{k \,|\, k \in \mathcal{V}, k' \in \mathcal{N}_t^{(i-1)}, (k, k') \in \mathcal{E}\}$, which extends the $(i-1)$-hop neighborhood by one additional hop. Hence, the academic graph is constructed as $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$, where $\mathcal{V}_t$ contains nodes in the multi-hop neighborhood of t, and $\mathcal{E}_t$ represents the edges between these nodes. Thus, an academic task is defined as follows:

$$f_\theta(\mathcal{G}_t) \to \mathbf{y}_t, \tag{1}$$

where f_θ represents a model with parameters θ. Furthermore, the academic tasks are broadly classified into predictive and generative tasks. If the label $\mathbf{y}_t$ is from a limited set of possible outcomes, this task is categorized as a predictive task; If the label $\mathbf{y}_t$ is in an open-ended output space, this task is categorized as a generative task. **For predictive tasks**, models (specified in Section 5.1) are considered as MLP-based, Embedding-based, GNN-based, or GWM-based, where the GWM framework efficiently integrates graph-structured data with LLM (Feng et al., 2025b). The training loss varies across different predictive tasks. **For generative tasks**, models are primarily based on LLMs. Supervised fine-tuning (SFT) is used for training, with the loss defined as follows:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{\sum_{t=1}^{T} L_t} \sum_{t=1}^{T} \sum_{i=1}^{L_t} \log p_\theta\big(y_{t,i} \,\big|\, y_{t,<i}, \mathcal{G}_t\big), \tag{2}$$

where L_t is the length of $\mathbf{y}_t = [y_{t,1}, ..., y_{t,L_t}]$, and $\log p$ is the log-likelihood. In this paper, six academic tasks are defined to demonstrate the four key features of RESEARCHARCADE. Table 1 summarizes the tasks under the two-step scheme with detailed task definitions in Appendix A.3.

4.2.1 ACADEMIC TASK 1: FIGURE/TABLE INSERTION

Proper placement of figures and tables evaluates models' ability to capture structural relationships and multi-modal content in academic papers. We formulate this as a multi-class classification task: given an academic graph $\mathcal{G}_t$ with all paragraphs, citations, figures, and tables, predict the paragraph $\hat{\mathbf{y}}_t$ associated with a target figure or table, where the ground truth $\mathbf{y}_t$ is the paragraph that contains or references it. Training uses contrastive cross-entropy loss (Chen et al., 2020) to minimize embedding distance between figures/tables and their corresponding paragraphs. For a given figure or table with embedding q_t and the embeddings of all paragraphs $\{p_i\}_{i=1}^N$, the loss is defined as:

$$\mathcal{L}_{\text{CE}}(\theta) = -\log \frac{\exp(\text{sim}(q_t, p_{\mathbf{y}_t})/\tau)}{\sum_{i=1}^{N} \exp(\text{sim}(q_t, p_i)/\tau)}, \tag{3}$$

where θ denotes the model parameters; $\mathrm{sim}(\cdot,\cdot)$ computes the cosine similarity between normalized embeddings; N is the total number of paragraphs; and τ is the temperature parameter that controls the sharpness of the probability distribution.

4.2.2 ACADEMIC TASK 2: PARAGRAPH GENERATION

Understanding how to generate specific paragraphs within their proper context is essential for both comprehending and writing academic papers. This generative task is defined as follows: given the input, an academic graph $\mathcal{G}_t$ including surrounding paragraphs, referenced figures and tables, and cited literature, generate the missing paragraph content $\hat{\mathbf{y}}_t$. The original paragraph content serves as the ground truth label $\mathbf{y}_t$. To train the LLM, SFT loss (Eq. 2) is utilized. The prompt designed to help the LLM better understand the document completion task is shown in Appendix A.4.1.

4.2.3 ACADEMIC TASK 3: REVISION RETRIEVAL

Identifying the precise location of revisions from reviewers' comments is essential for paper refinement. This captures intra-paper dynamics during peer review and demonstrates RESEARCHARCADE's ability to model evolving content. We formulate this as a top-k ranking task: given an academic graph $\mathcal{G}_t$ containing paper paragraphs and reviews, predict the top-k modified paragraphs $\hat{\mathbf{y}}_t$, with ground truth $\mathbf{y}_t$ denoting the actual revised paragraphs. Training employs the InfoNCE loss (He et al., 2020), which minimizes embedding distance between reviews and revised paragraphs while maximizing distance from unchanged ones:

$$\mathcal{L}_{\mathrm{InfoNCE}}(\theta) = -\frac{1}{R}\sum_{r=1}^{R}\log\frac{\sum_{i=1}^{M^+}\exp\left(\mathrm{sim}(q_r,k_i^+)/\tau\right)}{\sum_{i=1}^{M^+}\exp\left(\mathrm{sim}(q_r,k_i^+)/\tau\right)+\sum_{j=1}^{M^-}\exp\left(\mathrm{sim}(q_r,k_j^-)/\tau\right)}, \tag{4}$$

where θ denotes the model parameters; q_r is the model-generated embedding of the r-th review ($r = 1, ..., R$); k_i^+ and k_j^- are the embeddings of the i-th modified and j-th unchanged paragraph, respectively; M^+ and M^- are their counts; $\mathrm{sim}(\cdot,\cdot)$ is the similarity function; and τ is the temperature in the InfoNCE loss.

4.2.4 ACADEMIC TASK 4: REVISION GENERATION

Building on Section 4.2.3, this task focuses on generating quality-enhancing revisions of localized paragraphs conditioned on reviewer feedback, further demonstrating RESEARCHARCADE's dynamic evolution capability. Formally, given an academic graph $\mathcal{G}_t$ containing the original paragraph and its reviews, the goal is to generate a revised paragraph $\hat{\mathbf{y}}_t$, with the actual revision $\mathbf{y}_t$ as the label. Training uses SFT loss (Eq. 2), supported by a task-specific prompt (Appendix A.4.3) to guide the LLM in leveraging graph structures. Since LLMs have limited context length, reviews are first summarized using Qwen3-8B with the prompt in Appendix A.4.3.

4.2.5 ACADEMIC TASK 5: ACCEPTANCE PREDICTION

Predicting the acceptance of academic papers is a meaningful but challenging task. We fuse ArXiv's comprehensive multi-modal paper graph with OpenReview's ground-truth acceptance labels and temporal information to enable the task, reflecting RESEARCHARCADE's multi-source, multi-modal, and dynamically evolving nature. We design the task as a binary classification problem: given the input, an academic graph $\mathcal{G}_t$ containing papers from conferences in previous years and their corresponding paragraphs with figures and tables, predict the paper acceptance $\hat{\mathbf{y}}_t$ (Accept or Reject) for the future year. The real paper acceptance is the label $\mathbf{y}_t$. Binary cross-entropy loss is utilized as the training loss:

$$\mathcal{L}_{\mathrm{BCE}}(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\Big[\mathbf{y}_t\log\hat{\mathbf{y}}_t + (1-\mathbf{y}_t)\log(1-\hat{\mathbf{y}}_t)\Big]. \tag{5}$$

where θ represents the model's parameters and T the total number of papers.

4.2.6 ACADEMIC TASK 6: REBUTTAL GENERATION

Generating rebuttal responses to official reviews is critical, as response quality strongly influences paper acceptance. This task leverages textual and multi-modal information from ArXiv along with

Table 2: **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

official reviews from OpenReview. Formally, given an academic graph $\mathcal{G}_t$ containing the review and its related paragraphs with figures and tables from ArXiv, the goal is to generate the author's response $\hat{\mathbf{y}}_t$, with the true response $\mathbf{y}_t$ as the label. Training uses SFT loss (Eq. 2), guided by a task-specific prompt (Appendix A.4.5) to help the LLM capture graph structure and task requirements. To address token length limits, only the top-3 related paragraphs, selected via cosine similarity between review and paragraph embeddings using Qwen3-Embedding-0.6B, are included.

4.3 PROMISING NEW TASKS ENABLED BY RESEARCHARCADE

The versatility of RESEARCHARCADE extends beyond the tasks defined above, supporting additional stages of the research pipeline such as idea brainstorming, experiment planning, scientific writing, and peer reviewing—core activities in the academic process. These promising new tasks are illustrated in Figure 2, with detailed specifications provided in Appendix A.6.

5 EXPERIMENT

5.1 EXPERIMENT SETUP

Dataset: 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 information is provided in Appendix A.5.

Base Models: To demonstrate the compatibility of RESEARCHARCADE with diverse models, experiments are conducted across various base models.

(1) Embedding model (EMB): Considering the relatively long token input for our academic tasks, we utilize Longformer (Beltagy et al., 2020), a model designed for processing long documents.

(2) Graph neural network (GNN): Since the academic graphs constructed from our database are highly relational and heterogeneous, we consider HANConv (Wang et al., 2019), a heterogeneous graph attention neural network, as our GNN-based model.

(3) Large language model (LLM): We mainly leverage Qwen3-0.6B and Qwen3-8B (Team, 2025) as our LLM-based models, as they outperform models with an approximate number of parameters and are comparable to larger models in various evaluation tasks.

(4) Graph world model (GWM): To efficiently integrate graph-structured data with LLMs, we employ the embedding-based GWM (Feng et al., 2025b). It adopts a multi-hop aggregation to perform an embedding-level message passing, yielding an enhanced graph representation, which facilitates better LLM comprehension of the graph-structured data. Qwen3-0.6B (Team, 2025) is utilized as the LLM module for the GWM-based models.

Encoders: For the **text modality**, we represent text data as vector embeddings for integration with GNN-based and GWM-based models. Specifically, Longformer (Beltagy et al., 2020) is used for downstream GNNs, while Qwen3-Embedding-0.6B (Zhang et al., 2025) is adopted in GWM-based models to align with the Qwen3 LLM module. For the **visual modality**, LLaVA-1.5-7B (Liu et al.,

Table 3: **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)

Figure/Table Insertion				**Paragraph Generation**			
Model\Metric	Accuracy	AUC - ROC	MCC	Model\Metric	SBERT	Rouge-L	BLEU
EMB (Longformer, Fixed, 1-hop)	0.817	0.969	0.204	GWM (Qwen3-0.6B, Trained, 0-hop)	0.266	0.083	0.027
GNN (HANConv, Trained, 1-hop)	**0.880**	**0.977**	**0.296**	GWM (Qwen3-0.6B, Trained, 1-hop)	0.272	**0.086**	**0.028**
GNN (HANConv, Trained, 3-hop)	0.827	0.975	0.262	GWM (Qwen3-0.6B, Trained, 3-hop)	0.274	0.084	0.027
GNN (HANConv, Trained, 5-hop)	0.705	0.968	0.193	GWM (Qwen3-0.6B, Trained, 5-hop)	**0.276**	**0.086**	0.027
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)	**0.55**	**0.526**	**0.115**
GWM (Qwen3-0.6B, Trained, 1-hop)	0.304	0.325	0.264	GWM (Qwen3-0.6B, Trained, 1-hop)	0.47	0.478	-0.063
GWM (Qwen3-0.6B, Trained, 3-hop)	0.306	**0.326**	**0.265**	GWM (Qwen3-0.6B, Trained, 3-hop)	0.527	0.524	0.052
Revision Generation				**Rebuttal Generation**			
Model\Metric	SBERT	Rouge-L	BLEU	Model\Metric	SBERT	Rouge-L	BLEU
LLM (Qwen3-0.6B, Fixed, 1-hop)	0.321	0.210	0.147	LLM (Qwen3-0.6B, Fixed, 1-hop)	0.617	0.127	0.010
LLM (Qwen3-0.6B, Trained, 1-hop)	**0.733**	**0.554**	**0.468**	LLM (Qwen3-0.6B, Trained, 1-hop)	0.637	0.107	**0.026**
LLM (Qwen3-8B, Fixed, 1-hop)	0.704	0.446	0.276	LLM (Qwen3-8B, Fixed, 1-hop)	**0.717**	**0.164**	0.022

2024) converts figures into textual descriptions, which are then encoded using the same text encoders. Although we experimented with CLIP, our current approach is more effective and simpler to implement. The framework remains flexible and can accommodate alternative multi-modal encoders.

Evaluation Metrics: To systematically evaluate the performance of different models on our academic tasks, different evaluation metrics are considered for each task.

(1) Predictive Tasks: For the top-k ranking task, we report the top-5 precision, top-5 recall, and top-5 F-1 score to assess the model's performance. For the classification task, accuracy, AUC-ROC score, and Matthews correlation coefficient (MCC) are computed for evaluation.

(2) Generative Tasks: The semantic similarity between generated and reference answers is measured using the SBERT similarity score (Reimers & Gurevych, 2019). Lexical and n-gram overlap is assessed with Rouge-L (Lin, 2004) and BLEU (Papineni et al., 2002).

5.2 Experiment Results

The conclusive analysis of the experiment results is as follows, with a detailed analysis of each task provided in Appendix A.7.

5.2.1 ResearchArcade is General

Table 3 shows that ResearchArcade enables diverse tasks by integrating academic corpora with multi-modal information from ArXiv and peer reviews with revisions from OpenReview, while supporting various models by converting the data into CSV or JSON formats. EMB-based, GNN-based, and GWM-based models are capable of performing predictive tasks, while LLM-based models handle the generative tasks. Furthermore, the data quality in ResearchArcade is validated, with trained smaller LLMs approaching the performance of larger ones. In *Revision Generation*, Qwen3-0.6B's SBERT similarity score improves from 0.321 to 0.733, surpassing 0.704, the score of Qwen3-8B. And in *Rebuttal Generation*, Qwen3-0.6B's SBERT similarity score improves from 0.617 to 0.637, approaching 0.717, the score of Qwen3-8B.

5.2.2 ResearchArcade Models Dynamic Evolution

As shown in Table 3, ResearchArcade effectively captures dynamic evolution at both the intra-paper and inter-paper levels by incorporating temporal data from ArXiv and OpenReview. The tasks of *Revision Retrieval* and *Revision Generation* highlight ResearchArcade's ability to model

Table 4: **Ablation Study on multi-model 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, or without Figure & Table.

Rebuttal Generation				**Paragraph Generation**			
Model\Metric	SBERT	Rouge-L	BLEU	Model\Metric	SBERT	Rouge-L	BLEU
LLM (Qwen3-8B, Fixed, w/o F&T)	0.693	0.149	0.012	GWM (Qwen3-0.6B, Trained, w/o F&T)	0.259	0.078	0.025
LLM (Qwen3-0.6B, Fixed, w/o F&T)	0.558	0.105	0.005	GWM (Qwen3-0.6B, Trained, w F)	0.258	0.081	0.026
LLM (Qwen3-8B, Fixed, w F&T)	0.717	0.164	0.022	GWM (Qwen3-8B, Trained, w T)	0.255	0.080	0.023
LLM (Qwen3-0.6B, Fixed, w F&T)	0.617	0.127	0.010	GWM (Qwen3-0.6B, Trained, w F&T)	0.272	0.083	0.027

intra-paper evolution, predicting and generating revisions that reflect the continuous development of manuscripts. In particular, the top-5 F1 scores achieved by GNN-based and GWM-based models (0.265 each) outperform the EMB-based model (0.145), underscoring the framework's effectiveness. Importantly, incorporating OpenReview rebuttal data proved essential: before training, the model performed poorly, but after training, it was able to retrieve non-trivial revision paragraphs and produce meaningful manuscript revisions. In contrast, the *Acceptance Prediction* task reflects inter-paper evolution, aiming to identify promising papers for acceptance by learning from historical data. Here, performance was much poorer, with the best accuracy reaching only 0.55, barely above random chance. This emphasizes the inherent difficulty of predicting research trends.

5.2.3 Relational Graph Structure Delivers Consistent Gains

To assess the effectiveness of ResearchArcade's graph-centric design, we compare graph-based models (GNN-based and GWM-based) with non-graph models (EMB-based and MLP-based) across three tasks, observing performance gains of 7.7%, 67%, and 7.2% in *Figure/Table Insertion*, *Revision Retrieval*, and *Acceptance Prediction*, respectively, in Table 3. Multi-hop aggregation further improves performance, particularly in *Acceptance Prediction*: while 1-hop aggregation yields weak results (accuracies of 0.507 and 0.47), expanding to 3 hops raises both GNN-based and GWM-based models to 0.55, surpassing the MLP baseline (0.513). This indicates that acceptance decisions depend on higher-order context, such as venue affiliation and temporal trends, captured by multi-hop neighborhoods. However, for other tasks (e.g., *Revision Retrieval*, *Paragraph Generation*, *Figure/Table Insertion*), additional hops provide little benefit or even degrade performance. For instance, in *Figure/Table Insertion*, accuracy declines monotonically from 0.880 (1-hop) to 0.827 (3-hop) and sharply to 0.705 (5-hop), attributing to the sparsity of review and paper graphs centered on individual papers.

5.2.4 Multi-Modal Information Is Critical

Table 4 shows that incorporating figures and tables consistently enhances model performance compared to text-only baselines in both zero-shot and training settings for the *Rebuttal Generation* and *Generate Missing Paragraph* tasks. The inclusion of visual and tabular data augments the model's understanding of textual content, leading to clear performance gains. For the revision generation task, scores increase from 0.693 to 0.717 for the larger model and from 0.558 to 0.617 for the smaller model. Similarly, in the paragraph generation task, models benefit from the full modalities, improving from 0.259 to 0.272. These results validate ResearchArcade's multi-modal design and highlight the effectiveness of its approach to encoding multi-modal information.

6 Conclusion

We introduced ResearchArcade, a graph-based interface that unifies multi-source (ArXiv, OpenReview), multi-modal (text, figures, tables), and temporally evolving academic data into a coherent multi-table format. Building on a simple two-step scheme, (i) identify the target entity (label) and (ii) retrieve a task-specific academic graph (neighborhood), ResearchArcade standardizes the definition of both predictive and generative academic tasks. ResearchArcade is compatible with various models, serving as a valuable platform for studying research progress and developing models that facilitate automated scientific research.

Ethics Statement

We developed this work in accordance with the ICLR Code of Ethics and have carefully considered its broader impacts on the academic research community. Our system aims to contribute positively to research automation by providing tools for paper discovery, review assistance, and research trend analysis that could democratize access to academic insights and support researchers across different resource levels.

Potential Risks and Mitigation: We acknowledge several areas of concern regarding our academic task automation capabilities. Automated features such as paper completion and response drafting could potentially be misused for academic misconduct. We emphasize that our system is intended as a research assistance tool to augment human judgment, not replace academic thinking or writing. Additionally, our reliance on existing academic data sources (ArXiv, OpenReview) may perpetuate existing biases in publication patterns and review processes. The acceptance prediction capabilities could inadvertently influence submission strategies in ways that prioritize predicted acceptance over scientific merit rather than encouraging methodological rigor and novelty.

Data and Privacy: Our system uses exclusively publicly available academic data from ArXiv and OpenReview platforms. We respect the existing terms of use for these platforms and do not attempt to de-anonymize review processes or access private information. No human subjects are directly involved in our research process, and no additional ethical approvals were required.

Transparency and Responsible Use: We acknowledge that our graph construction and task formulation choices embed assumptions about academic workflows that may not generalize across all research domains. We encourage users to employ our system as an exploratory and assistance tool rather than for automated decision making, particularly for high-stakes academic decisions. Any research assistance provided should be subject to appropriate human oversight and verification to maintain research integrity.

Reproducibility Statement

To ensure reproducibility of our results, we have made extensive efforts to document our methodology and provide necessary resources. Complete implementation details for our graph construction process, including multi-source data integration from ArXiv and OpenReview, are provided in A.2.1 and A.2.2. The two-step task formulation scheme is fully specified in Section 4 with concrete examples. All experimental configurations, hyperparameters, and model architectures used across the six representative tasks are detailed in 5.1 and A.5. We provide comprehensive ablation studies and statistical significance testing procedures in 5.2. Code for data processing, graph construction, model implementation, and evaluation will be made available upon publication. The constructed heterogeneous graph dataset, along with task-specific splits and evaluation protocols, will also be released to facilitate future research.

REFERENCES

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Tao Feng, Yihang Sun, and Jiaxuan You. Grapheval: A lightweight graph-based llm framework for idea evaluation. *arXiv preprint arXiv:2503.12600*, 2025a.

Tao Feng, Yexin Wu, Guanyu Lin, and Jiaxuan You. Graph world model. *arXiv preprint arXiv:2507.10539*, 2025b.

Sujatha Das Gollapalli and Xiaoli Li. EMNLP versus ACL: Analyzing NLP research over time. In Lluís Màrquez, Chris Callison-Burch, and Jian Su (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2002–2006, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1235. URL https://aclanthology.org/D15-1235/.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

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. *arXiv preprint arXiv:1804.09635*, 2018.

Shengzhi Li and Nima Tajbakhsh. Scigraphqa: A large-scale synthetic multi-turn question-answering dataset for scientific graphs. *arXiv preprint arXiv:2308.03349*, 2023.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S Weld. S2orc: The semantic scholar open research corpus. *arXiv preprint arXiv:1911.02782*, 2019.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 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, June 2023. doi: 10.1109/jcdl57899.2023.00020. URL http://dx.doi.org/10.1109/JCDL57899.2023.00020.

Anirudh Sundar, Jin Xu, William Gay, Christopher Richardson, and Larry Heck. cpapers: A dataset of situated and multimodal interactive conversations in scientific papers. *Advances in Neural Information Processing Systems*, 37:66283–66304, 2024.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yang Tian, Zheng Lu, Mingqi Gao, Zheng Liu, and Bo Zhao. Mmcr: Benchmarking cross-source reasoning in scientific papers. *arXiv preprint arXiv:2503.16856*, 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. ISSN 1751-1577. doi: 10.1016/j.joi.2023.101399. URL https://doi.org/10.1016/j.joi.2023.101399.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pp. 2022–2032, 2019.

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. *arXiv preprint arXiv:2406.11633*, 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.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

A APPENDIX

A.1 DATA DESCRIPTION IN RESEARCHARCADE

The detailed dataset description is shown in Figure 3.

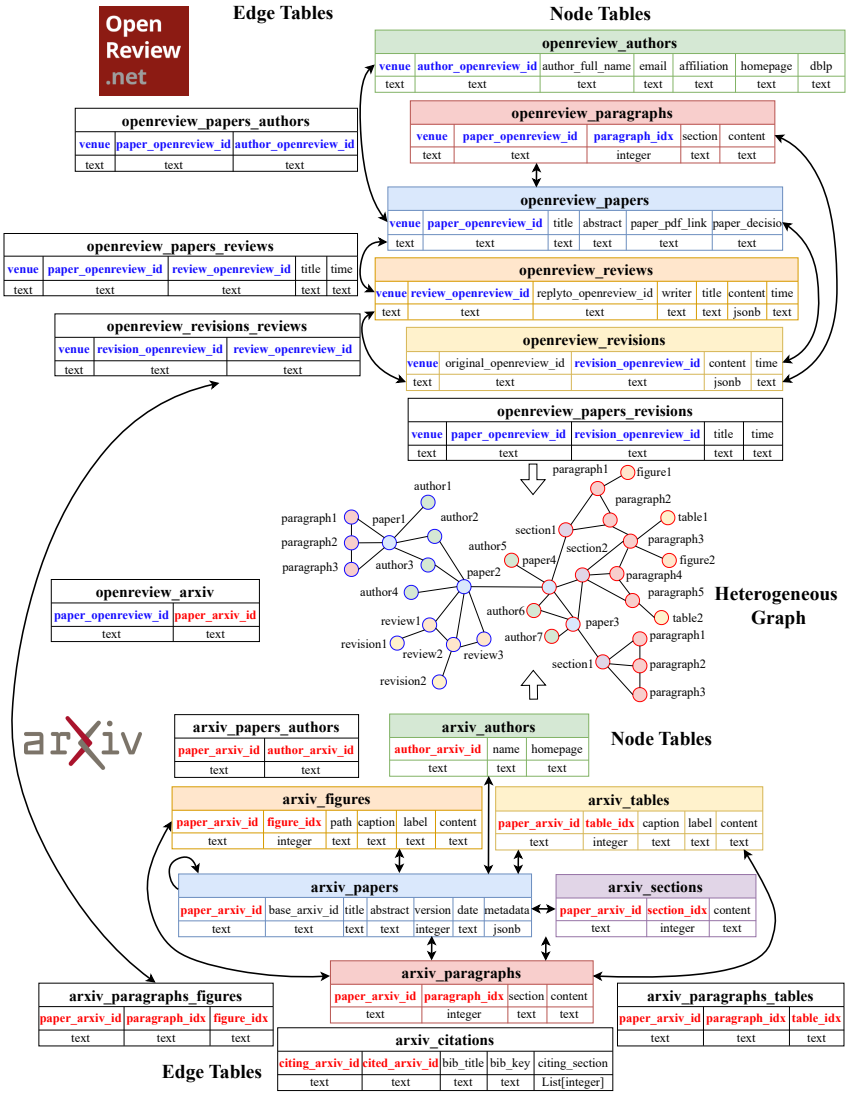

Figure 3: **A comprehensive overview of RESEARCHARCADE.** RESEARCHARCADE uses a multi-table format with graph structures to collect data from different sources with multiple modalities. Tables are classified into node tables (colored) or edge tables (black and white). The blue (denoting the OpenReview 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. The conversion from the multiple tables to heterogeneous graphs is straightforward.

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

The statistical overview of data collected from OpenReview is illustrated in Table 6.

The statistical overview of or_ArXiv table is shown in Table 7.

Category	#papers	#sections	#paragraphs	#figures	#tables	#authors
cs.AI	12018	110015	1425235	158906	69330	4814
cs.LG	16758	158907	2080025	270816	92389	4870
cs.CV	10236	81292	863198	145521	61551	4084
cs.RO	2632	19490	232443	35774	9996	1098
cs.CR	2056	18317	289377	24485	10900	689
cs.DB	353	2732	82988	4494	1445	184
cs.DC	934	7372	129567	12317	3097	427
cs.PF	158	1202	19460	2538	637	63
cs.MA	596	5139	81831	8113	2249	258
cs.OS	53	428	6085	1070	155	18
cs.other	19082	185809	2895718	226501	103044	6999
Non-CS	336	3604	49268	4681	887	0
Total	45794	405094	5210207	664034	261749	20505

Table 5: **Statistic overview of the data collected from ArXiv.** Note that overlaps exist among papers within computer science categories, since individual papers may be assigned multiple categorical classifications. The cs.other designation includes papers within computer science domains containing additional cs.xx subcategories beyond those enumerated above. Non-CS papers represent publications exclusively associated with non-computer science disciplinary categories.

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 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.

A.2 Data Collection Procedure

A.2.1 ArXiv

We developed a systematic pipeline to collect papers from ArXiv. The process begins by identifying target papers through either specific ArXiv IDs or publication date ranges. Using the ArXiv API, we download the LaTeX source code along with essential metadata including paper titles, authors, publication dates, and version information.

From the LaTeX source code, we extract key document elements and organize them into a graph structure. The initial graph contains nodes representing $\mathrm{sections}$, $\mathrm{figures}$, and tables, connected by edges that capture relationships such as $\mathrm{paper} - \mathrm{figure}$, $\mathrm{paper} - \mathrm{table}$, and $\mathrm{citation}$ links. We also integrate $\mathrm{paper} - \mathrm{category}$ relationships established during the collection phase.

To enable fine-grained analysis, we further decompose each paper by extracting individual paragraphs and adding them as $\mathrm{paragraph}$ nodes to our graph. This expansion creates additional relationship edges including $\mathrm{paragraph} - \mathrm{citations}$, $\mathrm{paragraph} - \mathrm{figures}$, and $\mathrm{paragraph} - \mathrm{tables}$, allowing for detailed content analysis and cross-referencing.

Using the author metadata collected initially, we enhance our database by creating dedicated author profiles through the Semantic Scholar API. By querying papers using their ArXiv IDs, we retrieve corresponding Semantic Scholar identifiers and homepage URLs when available. This process enables us to construct comprehensive author tables and establish $\mathrm{paper} - \mathrm{author}$ relationship mappings.

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 7: **Statistic overview of** openreview_arxiv **table.** 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.

While our initial graph construction captures citation information, many cited papers may not exist in our database, and some citations lack ArXiv IDs. To address these gaps, we use the ArXiv API to search for missing ArXiv identifiers of cited papers. Once identified, we download the LaTeX source code for these additional papers and integrate them into our graph using the same systematic approach, ensuring a more complete citation network.

A.2.2 OPENREVIEW

The detailed procedures used to collect and compile data from the OpenReview. An overview of the resulting dataset's content is provided in Figure 1.

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. Note that we do not collect the withdrawn papers. This step mainly contributes to the construction of the or_papers table and the 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 authors table.

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. The collected data is then used to form the or_reviews table and the or_papers_reviews table.

To construct the or_paragraphs table, we first download the PDF files and utilize pdfminer to extract the text from papers. The extracted text is then organized into paragraphs based on the distance between consecutive words. This table includes both the paragraphs of the papers and the paragraphs of their corresponding revisions.

For the or_revisions table and the or_papers_revisions table, we begin by retrieving the revision timestamps and PDF links for each submission via the OpenReview API. Since our focus is on the content of the revisions, we also download the PDFs of both the original and revised papers (the revised version is inferred based on the revision timestamp). The text is organized into paragraphs, as in the construction of the paragraphs table, and difflib is then employed to identify the differences between the original and revised texts. Finally, these differences are referred back to paragraphs.

Finally, to construct the or_revisions_reviews table, we assume that the current revision is created by discussions between the reviewers and authors, occurring between the time of the previous revision and that of the current revision. Thus, this table is constructed by leveraging the time information from the or_revisions table and or_reviews table.

A.3 EVALUATION TASK DEFINITIONS

A.3.1 FIGURE/TABLE INSERTION

Step 1: The target entity t is an arxiv_paragraph_figure or arxiv_paragraph_table edge, labeled with the indices of its parent paragraphs $\mathbf{y}_t$.

Step 2: The academic graph $\mathcal{G}_t$ is the full paper, containing arxiv_paper, arxiv_section, arxiv_paragraph, arxiv_figure, and arxiv_table nodes. Sections and paragraphs are sequentially linked and hierarchically connected with papers and figures/tables. arxiv_citation are included as external nodes linked to citing paragraphs.

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 4.2.5, which contains the relevant paragraphs with figures and tables.

A.4 PROMPT USAGE

A.4.1 GENERATE MISSING PARAGRAPHS

The following prompt is used for the GWM-based models.

```
{ 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.
```

Here, {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.4.2 REVISION RETRIEVAL

The following prompt is used for the GWM-based models.

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

Here, {review_graph} is an embedding-based token that is processed by multi-hop aggregation.

A.4.3 REVISION GENERATION

The following prompt is used to let LLM summarize the review.

```
REVIEW: {review}

INSTRUCTIONS:
- Summarize the following review into less than 150 words.
- Output only the summarization, enclosed between [START] and [END],
without any extra explanation or analysis.

OUTPUT: [START]your summarization here[END]
```

Here, {review} is the text-based content of a single review.

The following prompt is used to let LLM generate the revised paragraph based on the feedback from reviewers.

```
REVIEWS: {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]
```

Here, {review_graph} is the text-based token that sequentially connects the reviews. (e.g., Official Review by Reviewer, ...; Response by Authors: ...)

A.4.4 ACCEPTANCE PREDICTION

The following prompt is used for the GWM-based models.

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

Here, {paper_graph} is an embedding-based token that is processed by multi-hop aggregation.

A.4.5 REBUTTAL GENERATION

The following prompt is used to enable the LLM to generate the author's response based on the provided official review.

```
REFERENCES: {paper_graph}
QUESTIONS: {official_review}

INSTRUCTIONS:
- You are the author responding to the reviewer's comments.
- 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.

REVISED PARAGRAPH: [START]author's response here[END]
```

Here, {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.5 DATA USAGE IN EXPERIMENTS

A.5.1 FIGURE/TABLE INSERTION

In this task, we use 2,000 ar_paper nodes containing 15,535 ar_figure and ar_table nodes. Each paper also includes ar_section nodes and ar_citation edges. The ar_figure and ar_table nodes are split into 12,428 for training and 3,107 for testing.

A.5.2 PARAGRAPH GENERATION

In this task, we use 1,600 ar_paragraph nodes together with their connected ar_figure, ar_table, and ar_citation nodes and edges, with 1,280 allocated for training and 360 for testing.

A.5.3 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.5.4 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.5.5 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.5.6 REBUTTAL GENERATION

Using 3,077 $\mathrm{or\_paper}$ nodes that has connected with a $\mathrm{ar\_paper}$ node based on $\mathrm{or\_ArXiv}$ table from ICLR 2025—split 2779/298 into train/test—yields 1,239 and 11,620 authors' responses, with node type $\mathrm{or\_review}$, for training and testing, respectively.

A.6 PROMISING NEW TASKS

In this part, we list out and describe what tasks can be performed on RESEARCHARCADE in each research stage.

A.6.1 IDEA GENERATION

Brainstorming research ideas based on existing works is an essential skill for any researcher. Enhancing 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.6.2 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.6.3 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.6.4 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.7 EXPERIMENT RESULTS ANALYSIS

A.7.1 FIGURE/TABLE INSERTION

In this subsection, we present the evaluation results for figure and table insertion.

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

Experimental Results. Table 3 highlights two key findings: (1) adjacent neighborhoods provide sufficient information, significantly improving prediction accuracy; and (2) incorporating larger neighborhoods leads to performance fluctuations and even degradation. This may be attributed to the relative sparsity of the paper graph and the incomplete collection of citation, figure, or table information during the paper processing stage.

A.7.2 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 tested multi-modal inputs using four conditions: both figures and tables, figures only, tables only, and neither component.

Experiment Results Table 3 shows two key findings for neighborhood information: (1) models achieve optimal performance when provided with the most comprehensive neighborhood data, and (2) excluding neighborhood information significantly degrades performance.

For multi-modal information, the results demonstrate that (1) complete multi-modal data (both figures and tables) yields the best performance, and (2) partial multi-modal information performs no better than providing no multi-modal data at all.

A.7.3 REVISION RETRIEVAL

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

Experimental Results. From Table 3 we can observe that: (1) Models optimized with InfoNCE (GNN-/GWM-based) 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.7.4 REVISION GENERATION

Baselines. For the generative task, we use LLM-based models as baselines. Qwen3-8B is 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 3, with the following observations: (1) A substantial performance gap exists between zero-shot Qwen3-0.6B and Qwen3-8B, which is reasonable in view of their different sizes of parameters. (2) After supervised fine-tuning Qwen3-0.6B, its performance was significantly enhanced, approaching the zero-shot performance of Qwen3-8B. These highlight the effectiveness of RESEARCHARCADE in facilitating LLMs' understanding of the dynamic evolution within a paper.

A.7.5 ACCEPTANCE PREDICTION

Baselines. MLP-based, GNN-based, and GWM-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 GWM-based models.

Experimental Results. As shown in Table 3, 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, GWM-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 GWM-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.7.6 REBUTTAL GENERATION

Baselines. In the generative setting, LLM-based models are adopted as our baselines. Specifically, Qwen3-8B is 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 3 reveal the following insights: (1) There exists a performance gap between Qwen3-0.6B and Qwen3-8B models 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.

A.8 LLM USAGE DISCLOSURE

We used large language models (LLMs) to assist with literature search and identification of related work relevant to our research on graph-based academic data interfaces. Specifically, we employed LLMs to help discover papers across different research areas that intersect with our work, including graph neural networks, large language models, academic data mining, and research automation. All identified papers were subsequently verified by the authors, and we take full responsibility for the accuracy and appropriateness of all citations and related work discussions presented in this paper.

We also utilized LLMs to assist with paper writing, including improving grammar, enhancing clarity of explanations, and refining the presentation of our methodology and results. The LLMs were used as writing assistants to help articulate our ideas more clearly, but all technical content, experimental design, analysis, and conclusions remain the original intellectual contribution of the authors. We maintain full responsibility for all claims, representations, and technical content presented in this work, and have thoroughly verified all LLM-assisted content for accuracy and appropriateness.