

GraphReview: Scientific Paper Evaluation via LLM-based Graph Evidence Expansion

Pujun Zheng¹, Wanying Ren¹, Jiacheng Yao¹, Guoxiu He¹*, Star X. Zhao²

¹School of Economics and Management, East China Normal University

²Institute of Big Data, Fudan University

{pjzheng, wyren, jcyao}@stu.ecnu.edu.cn, gxhe@fem.ecnu.edu.cn

Abstract

Scientific paper evaluation often involves not only assessing a manuscript itself, but also relating it to contemporaneous research and prior literature. However, existing LLM-based methods typically model these signals separately and lack a unified mechanism for aggregating review evidence across papers. We propose **GraphReview**, a graph-based LLM framework that formulates paper evaluation as inference-time graph evidence expansion over a semantic paper graph. The graph jointly captures intrinsic quality, synchronic links among contemporaneous papers, and diachronic links to prior work. LLMs are used to estimate node-level quality priors and generate edge-level comparative evidence through pairwise paper comparisons, while Personalized PageRank integrates these structured signals for quality ranking, decision prediction, and review generation. To produce higher-quality graph evidence, we propose reward-induced maximum likelihood objectives for training the LLM backbones. Experiments show that GraphReview consistently outperforms the strongest baseline, achieving average improvements of 29.7% on decision and ranking metrics, including gains of 23.7% in Accuracy and 57.6% in Spearman’s ρ . It also produces higher-quality review texts and generalizes effectively across time periods and conference venues.

Introduction

Peer review plays a foundational role in ensuring publication quality (Alberts, Hanson, and Kelner 2008). As the volume of scholarly literature grows rapidly (He, Sun, and Lu 2023), leveraging artificial intelligence to assist paper reviewing has become an important strategy for addressing the reviewing crisis (Zhou, Chen, and Yu 2024; Du et al. 2024; Zhuang et al. 2025; Li et al. 2025). In particular, large language model (LLM)-based approaches have demonstrated remarkable capability and scalability in paper reviewing, and are increasingly adopted to support paper evaluations and help authors improve manuscript quality (Latona et al. 2024; Thakkar et al. 2025, 2026).

Despite recent progress, existing LLM-based methods for paper evaluation remain limited by narrow modeling assumptions, as shown in Figure 1 (top). The first line of work focuses on detailed analysis of textual content (Lu et al. 2024; Jin et al. 2024; Weng et al. 2025; Zhu et al. 2025; Zeng et al. 2025; Chang et al. 2025), treating paper quality as an intrinsic property of the manuscript. The second line com-

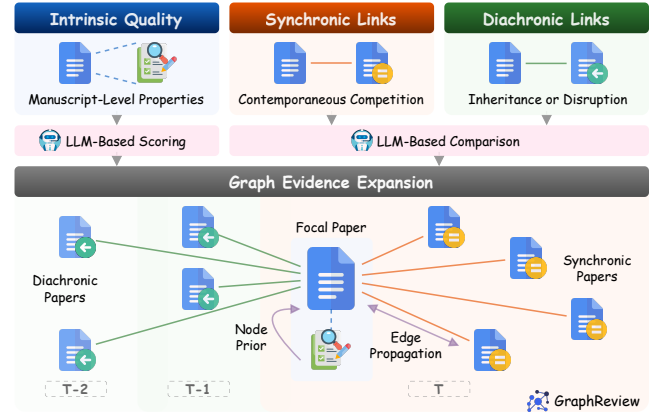


Figure 1: Previous LLM-based methods consider information sources in isolation (top), whereas GraphReview organizes multi-source review evidence in a semantic graph and aggregates it through inference-time expansion (bottom).

pares submissions within the same review cycle (Zhang et al. 2025; Zhao et al. 2025a; Zheng et al. 2026a), defining quality in terms of contemporaneous competition. The third line evaluates new papers through their relationships to prior literature (He et al. 2023; Xue et al. 2024; Zhao et al. 2025b), thus grounding quality in a historical scholarly context while emphasizing inheritance, continuation, and disruption.

However, existing methods typically treat different sources of review evidence separately, without a unified mechanism for integrating signals across papers. From a science-of-science perspective, evaluating a paper requires not only assessing its writing quality and scientific merit, but also situating it within contemporaneous competition and the evolving landscape of domain knowledge (Fortunato et al. 2018; Wu, Wang, and Evans 2019). These signals are central to real-world peer review, especially in batch-style review scenarios where submissions are assessed comparatively, yet methods based on a single perspective capture only part of the picture. Existing LLM-based approaches to paper evaluation primarily rely on end-to-end autoregressive generation or agent-based frameworks, which further exacerbates this limitation. Even when additional papers are incorporated, they are typically appended as unstructured context rather than modeled as explicit comparative evidence that can be propagated across related work. As a result, these methods

*Corresponding author.

struggle to capture comparative signals from the broader scientific context. This limitation motivates a graph-based framework that uses the graph as a symbolic inference structure for inference-time expansion, enabling context-aware modeling of multi-source review signals and leveraging their complementarity.

To this end, we propose GraphReview, a graph-based LLM framework that represents paper evaluation as inference-time evidence expansion over multi-source review signals within a semantic paper graph. As illustrated in Figure 1 (bottom), **Intrinsic Quality** captures manuscript-level properties such as originality, clarity, and significance; **Synchronic Links** model comparative relations among contemporaneous submissions; and **Diachronic Links** characterize how a paper extends, diverges from, or overlaps with recent literature. To operationalize this framework, GraphReview uses the graph to organize LLM-generated review signals rather than to learn a latent graph representation. The LLM first assigns each paper node a quality prior and then performs pairwise comparisons on selected edges to produce relational evidence. Personalized PageRank aggregates the pointwise priors and edge-level comparisons into a transductive global ranking and acceptance prediction under the target capacity, while review texts are generated in parallel from the same structured evidence. We further use reward-induced maximum likelihood as a stable soft-target objective to train the LLM signal generators.

We conduct comprehensive experiments from multiple perspectives to systematically evaluate the proposed framework. Results show that our method achieves leading overall performance, yielding an average relative gain of 29.7% over the strongest baseline on decision and ranking metrics. Specifically, it improves Accuracy by 23.7% and Spearman’s ρ by 57.6%. For review text quality, our method achieves win rates above 50% against all baselines in technical depth, evidence grounding, scientific rigor, revision utility, and overall preference. Furthermore, cross-time-period and cross-venue generalization studies show that the model reliably distinguishes papers at different quality levels. Our main contributions are as follows:

- We formulate paper evaluation in an integrated framework that captures intrinsic quality together with synchronic and diachronic relations.
- We propose an inference-time graph evidence expansion mechanism tailored to paper evaluation, and design training objectives suitable for its LLM backbones.
- Our method achieves leading performance on ranking and decision prediction, improves review text quality, and demonstrates robust generalization across time periods and conference venues.

Related Work

Automated Scientific Paper Evaluation

A major line of research evaluates the intrinsic quality of a paper by using LLM-based review agents to simulate peer review and produce automated pointwise scores (Jin et al. 2024; Lu et al. 2024). To improve review quality, many studies introduce iterative refinement or multi-round interaction

(Weng et al. 2025; Tan et al. 2024), domain-specific fine-tuning, reinforcement learning, and structured agent workflows (Zeng et al. 2025; Yu et al. 2024; Tyser et al. 2024; Garg et al. 2025; Chang et al. 2025; Zhu et al. 2025; Lu et al. 2025). Another line of work compares submissions within the same venue and adopts comparison-based evaluation. Representative directions include pairwise comparison for quality prediction (Zhao et al. 2025a; Höpner et al. 2025) and preference aggregation via reinforcement learning with comparative rewards (Zhang et al. 2025; Zheng et al. 2026a,b). A further line of research considers scholarly inheritance, which is important for assessing a paper’s actual contribution. This information has been explored in graph-based impact prediction (He et al. 2023; Xue et al. 2024), but only a few LLM-based review systems incorporate paper retrieval or listwise ranking (Zhao et al. 2025b; Zhu et al. 2025). Overall, most existing approaches rely on a single dominant perspective and do not combine intrinsic and relational signals within an integrated framework. Their autoregressive or agent-based architectures are also not well suited to explicitly organize multi-source relations, aggregate evaluative evidence, and support global decision making.

LLM-Based Graph Methods

Classic message-passing GNNs learn node representations and perform downstream predictions through neighborhood aggregation (Kipf and Welling 2016; Veličković et al. 2017; Gilmer et al. 2017; Hamilton, Ying, and Leskovec 2017). With the rapid development of LLMs, graph methods have become an increasingly important complementary component. Early work mainly converted graph structures into natural-language descriptions through hand-crafted rules (Wang et al. 2023; Zhao et al. 2023). More recent studies pursue tighter alignment between graph structures and linguistic semantics, enabling LLMs to better understand graphs and generalize across reasoning tasks (Tang et al. 2024; Wang et al. 2025; Jing et al. 2026; Luo et al. 2025; Tao et al. 2025; Feng, Sun, and You 2025). Another major direction uses graphs for retrieval-augmented generation, treating knowledge graphs as external memory and combining their structure with retrieval (Edge et al. 2024; Guo et al. 2024; Gutiérrez et al. 2024; Dong et al. 2025). Other work models graphs as interactive environments for decision making (Zhang 2023; Finkelshtein et al. 2025). However, existing LLM-based graph methods primarily use graph structures as context, external memory, or an interaction environment, rather than as a mechanism for organizing and aggregating quality evidence in evaluation tasks. As a result, they cannot explicitly represent high-level review signals, directional relationships, or evaluative evidence, making them less suitable for graph-based paper evaluation.

Methodology

Overview

Scientific paper evaluation is formulated as a batch-style transductive ranking problem. Given the whole set of n concurrently submitted papers, a known review capacity or acceptance rate γ , and an auxiliary corpus consisting

Algorithm 1: Inference-Time Graph Evidence Expansion

Require: Minimum improvement threshold ϵ and maximum patience $c_{\max}$

Ensure: Best round T^* and best ranking $\mathbf{r}^*$

```

1: Initialize  $\eta^* \leftarrow 0, T^* \leftarrow 1, \mathbf{r}^* \leftarrow \emptyset, T \leftarrow 1, c \leftarrow 0$ 
2: while  $c < c_{\max}$  do
3:   Construct adjacency matrix  $\mathbf{C}^{(T)}$  by  $f_{\text{S2FM}}$ 
4:   for each node  $v$  do
5:     for each  $u$  satisfying  $c_{uv}^{(T)} = 1$  do
6:       // Preference generation
7:        $g_{v \leftarrow u} = f_{\text{LLM}}(x_u, x_v, p_c)$ 
8:     end for
9:     // Evidence collection
10:     $\mathbf{g}_v = \bigoplus_{c_{uv}^{(T)} = 1} g_{v \leftarrow u}$ 
11:   end for
12:   // Ranking aggregation
13:    $\boldsymbol{\pi} = f_{\text{PPR}}(\{f_{\text{LLM}}(x_v, p_s)\}, \{\mathbf{g}_v\})$ 
14:    $\mathbf{r} = \text{argsort}(\boldsymbol{\pi})$ 
15:   Calculate the performance  $\eta$  from  $\mathbf{r}$ 
16:   if  $\eta - \eta^* > \epsilon$  then
17:      $\eta^* \leftarrow \eta, T^* \leftarrow T, \mathbf{r}^* \leftarrow \mathbf{r}, c \leftarrow 0$ 
18:   else
19:      $c \leftarrow c + 1$ 
20:   end if
21:    $T \leftarrow T + 1$ 
22: end while
23: return  $T^*, \mathbf{r}^*$ 

```

of $(N - n)$ relevant historical papers, the system takes $\mathbf{x} = [x_1, x_2, \dots, x_n, x_{n+1}, \dots, x_N]$ as input and produces a quality-based ranking of the submitted papers, denoted by $\mathbf{r} = [r_1, r_2, \dots, r_n]$. The final decision selects the top $\lfloor \gamma n \rfloor$ papers from this batch, rather than evaluating each paper as an isolated standalone instance. The overall process of our method is shown in Figure 2.

Graph Evidence Expansion

From a graph perspective, a Transformer input sequence with global attention can be viewed as a fully connected graph (Joshi 2025), and attention can be interpreted as a form of GNN computation (Frasca et al. 2025). Although inspired by this analogy, our method differs from conventional GNNs: the graph acts as a symbolic inference structure rather than a learned latent-representation model. Specifically, we formulate paper evaluation as graph-structured evidence expansion, where task-specific pairwise judgments are produced directly by LLMs. As summarized in Algorithm 1, inference consists of three message-passing-style stages: preference generation, evidence collection, and ranking aggregation.

We perform inference over a dynamically expanded graph that combines intrinsic quality, synchronic links, and diachronic links. Across rounds, S2FM activates semantic edges, the LLM generates edge-level pairwise evidence, and PPR aggregates node priors with relational evidence. Increasing T thus expands structured inference-time evidence, rather than adding recurrent GNN-style state updates, leading to improved and eventually convergent performance.

Preference Generation The edge-evidence function characterizes how a neighboring paper provides comparative evidence for a target paper. Unlike conventional GNNs that propagate learned embeddings or recurrent node states, our method uses the LLM to perform pairwise comparisons over selected edges. Given the textual content x_u and x_v of nodes u and v , we query the LLM f_{LLM} with a comparison prompt p_c to infer their relative quality. Each comparison returns two parts: a preference label, and a natural language rationale retained in g_v for text consolidation. Graph connectivity is specified by a dynamically expanded adjacency matrix $\mathbf{C}^{(T)}$. Each entry $c_{uv}^{(T)}$ indicates whether the semantic link between nodes u and v is activated by the Sequential 2-Factor Matching (S2FM) algorithm, denoted by f_{S2FM} . Since evidence is generated for both comparison directions, the adjacency matrix is symmetric, that is, $c_{uv}^{(T)} = c_{vu}^{(T)}$.

Evidence Collection The collection function gathers all edge-level evidence associated with a target node. Instead of compressing incoming signals with a pooling operator, we retain the full set of neighborhood evidence through concatenation. Let $\mathcal{N}(v)$ denote the neighbor set of node v . The aggregated evidence for node v is formed by concatenating the comparison results from all neighbors in $\mathcal{N}(v)$.

Ranking Aggregation The ranking aggregation step combines pointwise and relational evidence to produce the final ranking signal. Rather than training an end-to-end graph predictor, we integrate LLM-based pointwise priors with relational evidence through Personalized PageRank (PPR) (Jeh and Widom 2003), denoted by f_{PPR} , which yields a global ranking without training an entire graph model. For each node v , we query the LLM f_{LLM} with the prompt p_s to obtain a direct quality estimate for x_v . This score serves as a prior importance signal that captures standalone evidence for the paper. We then refine it with comparative evidence collected from neighboring nodes. Specifically, PPR diffuses node importance over the graph based on both the pointwise priors and the aggregated edge evidence. Finally, we apply argsort to the converged PPR scores $\boldsymbol{\pi}$ to obtain a total ranking, and predict the top $\lfloor \gamma n \rfloor$ papers as accepted under a preset acceptance rate γ , with the remaining papers rejected.

Graph Building and Computing

This section describes how we design f_{S2FM} to construct graphs and use f_{PPR} to perform computation on them.

Sequential 2-Factor Matching S2FM progressively expands the graph through repeated 2-factor matching steps, maintaining three desirable properties: scaling consistency, permutation equivariance, and guaranteed connectivity. Let $\mathbf{C}^{(t)} \in \mathbb{R}^{N \times N}$ denote the matching matrix at iteration t , where $c_{uv}^{(t)}$ indicates whether an edge exists between nodes u and v . Each iteration adds N bidirectional edges. At iteration t , S2FM solves a maximum-weight 2-factor matching problem, selecting a binary symmetric increment matrix $\Delta \mathbf{C}^{(t)}$ that maximizes the total semantic similarity $\sum_{u,v} \mathbf{h}_u^\top \mathbf{h}_v \Delta c_{uv}^{(t)}$ while enforcing that each node has degree 2. To avoid repeated edges across iterations, the constraint

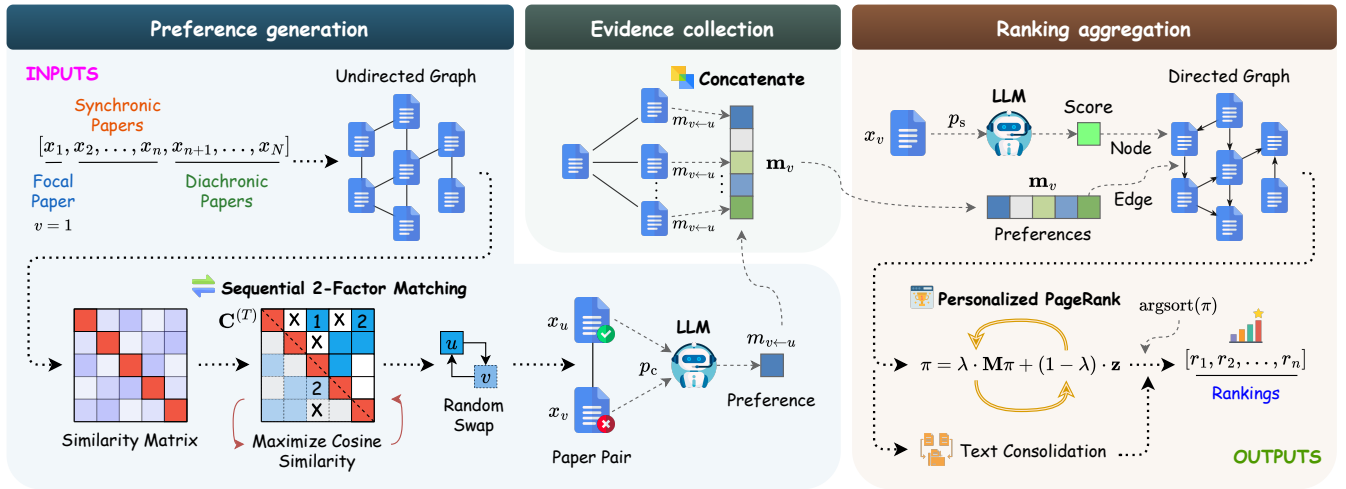


Figure 2: The overall process of our method. It includes preference generation (left), evidence collection (center), and ranking aggregation (right). The input is a list of papers, and the output is the ranking of the papers.

$\Delta c_{uv}^{(t)} \leq 1 - c_{uv}^{(t-1)}$ ensures that newly selected edges connect only previously unmatched node pairs, after which the adjacency matrix is updated as $\mathbf{C}^{(t)} = \mathbf{C}^{(t-1)} + \Delta \mathbf{C}^{(t)}$. After T iterations, S2FM produces the final adjacency matrix $\mathbf{C}^{(T)}$.

Personalized PageRank We introduce PPR to unify node-level pointwise predictions and edge-level pairwise preference constraints produced by the aggregation function. PPR performs a weighted random walk between the pointwise prior and local transition probabilities, yielding a global ranking score. For node-level evaluation, we first construct a normalized node prior $\mathbf{z} = [z_1, z_2, \dots, z_N]$ from pointwise prediction scores $\mathbf{e} = [e_1, e_2, \dots, e_N]$. We use a small constant ϵ to handle potentially non-positive scores or missing values, thereby ensuring a valid probability distribution:

$$z_u = \frac{\max(e_u, \epsilon)}{\sum_{k=1}^N \max(e_k, \epsilon)} \quad (1)$$

For edge-level comparisons, we initialize a graph with the bidirectional adjacency matrix $\mathbf{C} \leftarrow \mathbf{C}^{(T)}$. For each edge (x_u, x_v) , if f_{LLM} judges that x_u strictly outperforms x_v , we remove the directed edge from u to v by setting:

$$c_{uv} \leftarrow 0, \quad \text{if } x_u \succ x_v \quad (2)$$

The resulting adjacency matrix $\mathbf{C}$ represents a directed graph. Let $\mathbf{D} \in \mathbb{R}^{N \times N}$ be the diagonal out-degree matrix with $d_{vv} = \sum_{k=1}^N c_{vk}$. To construct a valid Markov chain, we define a column-normalized transition matrix $\mathbf{M} \in \mathbb{R}^{N \times N}$. For nodes with zero out-degree, we redirect according to the prior $\mathbf{z}$. Define each element m_{uv} as:

$$m_{uv} = \begin{cases} \frac{c_{vu}}{d_{vv}} & \text{if } d_{vv} > 0 \\ z_u & \text{if } d_{vv} = 0 \end{cases} \quad (3)$$

The final aggregated score vector $\boldsymbol{\pi} \in \mathbb{R}^N$ is the stationary distribution of this random walk, combining local pairwise transitions $\mathbf{M}$ with the global pointwise prior $\mathbf{z}$ via:

$$\boldsymbol{\pi} = \lambda \cdot \mathbf{M} \boldsymbol{\pi} + (1 - \lambda) \cdot \mathbf{z} \quad (4)$$

Here, $\lambda \in (0, 1)$ is the damping factor. We approximate $\boldsymbol{\pi}$ via power iteration.

In addition, since PPR cannot directly propagate or update textual rationales, we run a parallel text consolidation pipeline that merges node and edge explanations alongside PPR’s aggregation of numerical signals.

LLM Backbones

This section describes the training method for the LLM backbones used to generate node-level priors and edge-level comparative evidence, denoted by f_{LLM} .

Cold Start and Prompt Optimization To encourage the model to learn not only superficial response patterns but also the underlying evaluation criteria, we first perform cold-start supervised fine-tuning. Paper evaluation, however, is a specialized and cognitively demanding task, and even human experts may struggle to articulate criteria that are both effective and comprehensive. To construct prompts that cover the major dimensions of paper assessment, we adopt a simple self-optimization procedure that iteratively refines the prompt, and then use the optimized prompt to generate training responses. Training on these cold-start data enables the model not only to produce a special review-signal token representing the evaluation outcome, but also to explain the evaluation dimensions and key considerations that lead to the same conclusion.

Reward-Induced Maximum Likelihood After the cold-start stage, we use Reward-Induced Maximum Likelihood (RIML) as a simple and stable soft-target objective for aligning the LLM signal generators. Unlike rollout-based reinforcement learning, RIML does not require a costly reward pipeline or long-horizon exploration. It converts expert score rewards and pairwise preferences into target distributions over review-signal token classes, and then trains the cold-started model with maximum likelihood. This formulation better matches paper evaluation, where supervision comes from human rating distributions and comparative judgments

rather than verifiable multi-step rewards. Specifically, the objectives are defined as follows.

At the node level, the model predicts a distribution over discrete score anchors $\{a_k\}_{k=1}^K$. For a paper v with score s_v , we define:

$$\begin{aligned} P(k | q_v) &= \frac{\exp(z_{vk})}{\sum_{l=1}^K \exp(z_{vl})}, \\ y_{vk} &= \frac{\exp(r_{vk})}{\sum_{l=1}^K \exp(r_{vl})}, \quad r_{vk} = -\frac{(s_v - a_k)^2}{2\sigma^2} \quad (5) \\ \mathcal{L}_{\text{node}}(\theta) &= -\frac{1}{N} \sum_{v=1}^N \sum_{k=1}^K w_v y_{vk} \log P(k | q_v) \end{aligned}$$

Here, the distance-based reward induces a soft target over score anchors, preserving proximity in the continuous score space; we set $w_v = 1$ for all node samples.

At the edge level, for a pair (u, v) with sufficiently different ground-truth scores, we define:

$$\begin{aligned} P(k | q_{uv}) &= \frac{\exp(z_{uvk})}{\exp(z_{uv0}) + \exp(z_{uv1})}, \\ y_{uvk} &= r_{uvk}, \quad r_{uvk} = \mathbb{I}[k = \mathbb{I}[s_u - s_v > 0]], \quad (6) \\ \mathcal{L}_{\text{edge}}(\theta) &= -\frac{1}{M} \sum_{m=1}^M \sum_{k=0}^1 w_{uv} y_{uvk} \log P(k | q_{uv}) \end{aligned}$$

Here, $w_{uv} = |s_u - s_v|$ weights clearer pairwise preferences more strongly, and each valid pair corresponds to one edge index m .

Experiments

Experimental Setup

Dataset Construction All data are collected via the official OpenReview API at <https://openreview.net>. We construct the primary training and test sets from ICLR 2025 submissions, using the average human reviewer score as the ground-truth label. The external auxiliary knowledge base consists of all accepted papers from ICLR, ICML, and NeurIPS in 2023 and 2024, covering recent advances and the research frontier of the field. For each batch, we create a graph with N 1.5k nodes for n 0.5k submissions. This takes about 4 hours on two RTX Pro 6000 GPUs. To assess generalization across time periods and venues, we further sample papers from ICLR 2026 and ICML 2025. These data include expert-provided paper group annotations and span multiple quality levels.

Training Setup and Hyperparameters We use the earlier-released Qwen2.5-7B-Instruct (Qwen Team 2024) as the backbone model to reduce potential data leakage, and adopt LoRA (Hu et al. 2022) for efficient training. The acceptance rate γ is fixed at 31.4%, which is the average acceptance rate for ICLR 2023 and 2024.

Baselines We evaluate the following categories of baselines: **Classic GNNs**, including traditional GNN-based evaluation methods such as GCN (Kipf and Welling 2016), GAT (Veličković et al. 2017), and GraphSAGE (Hamilton,

Ying, and Leskovec 2017). **LLMs**, including direct prompting approaches that ask LLMs from multiple providers to evaluate papers, with the main results reported for earlier-released models such as GPT-5-Mini, Gemini-2.5-Flash, and DeepSeek-V3.2 to reduce the risk of data leakage. **Review Agents**, including agent-based evaluation systems such as AIScientist (Lu et al. 2024), AgentReview (Jin et al. 2024), SEA-7B (Yu et al. 2024), CycleReviewer-7B (Weng et al. 2025), DeepReview-7B, and DeepReview-14B (Zhu et al. 2025). **Comparative Systems**, including comparison-based evaluation methods such as PairReview (Zhang et al. 2025), CNPE-7B (Zheng et al. 2026a), NAIP-8B (Zhao et al. 2025b), and NAIPv2-8B (Zhao et al. 2025a).

Evaluation Metrics We use two groups of metrics to evaluate decision accuracy and ranking quality. The first formulates paper evaluation as a binary classification task that predicts whether a paper should be accepted, and we report Accuracy, F1, and AUC. The second measures the ability to rank high-quality papers ahead of low-quality ones, using Spearman’s ρ , Kendall’s τ , and NDCG@10. We calculate 95% confidence intervals and the statistical significance of differences using the bootstrap method.

Main Results

As shown in Table 1, classical GNN-based methods perform poorly overall. Although they are effective at modeling multi-source information with graph structures, they cannot generate textual feedback, which limits their practical value for reviewers and authors. More importantly, their architectures are not well suited to capturing the fine-grained semantics required for paper evaluation. In contrast, naïve LLMs achieve relatively strong performance on several metrics, consistent with their direct scoring paradigm that primarily relies on internal signals such as Intrinsic Quality. However, their reliability is undermined by the substantial risk of data leakage. The other two categories, review agents and comparison-based systems, are more competitive. Among review agents, DeepReview-14B trained with reinforcement learning performs best, achieving an average score of 0.5540 ($CI = [0.4950, 0.6136]$). Comparison-based methods place greater emphasis on modeling relational dependencies among contemporary papers and across time. NAIPv2 provides the strongest baseline performance, with an average score of 0.6087 ($CI = [0.5528, 0.6673]$).

Our GraphReview framework achieves an average score of 0.7893 ($CI = [0.7476, 0.8306]$) and consistently outperforms all baselines on both decision and ranking metrics. Compared with the strongest baseline, NAIPv2, our method achieves an average relative improvement of 29.7% ($p < 0.001$). On the core metrics, Accuracy and Spearman’s ρ , the relative gains reach 23.7% and 57.6% (both $p < 0.001$), respectively. These results demonstrate that our model is substantially more capable of both distinguishing accepted papers from rejected ones and predicting their relative ranking. Overall, the results validate the effectiveness of our framework, which is the only system that jointly integrates all information sources relevant to paper evaluation while also generating an advisory review for each paper.

Method	Method Characteristics					Decision Performance			Ranking Performance			Avg. Per- formance
	Intrinsic Quality	Syn- chronic Links	Dia- chronic Links	Review Text	Data Leakage Risk	Accuracy	F1	AUC	Spearman	Kendall	NDCG	
									ρ	τ	@10	
Classic GNNs												
GCN	✓	✓	✓		○	0.5980	0.5404	0.5720	0.1267	0.0852	0.5552	0.4129
GAT	✓	✓	✓		○	0.5760	0.5235	0.5104	0.0485	0.0349	0.5734	0.3778
GraphSAGE	✓	✓	✓		○	0.5640	0.5143	0.5055	0.0492	0.0348	0.5868	0.3758
LLMs												
GPT-5-Mini	✓			✓	●	0.7100	0.6605	0.7104	0.4136	<u>0.3317</u>	0.6982	0.5874
Gemini-2.5-Flash	✓			✓	●	0.6060	0.5387	0.5557	0.1936	0.1569	0.6422	0.4488
Deepseek-V3.2	✓			✓	●	0.6180	0.5527	0.5975	0.3536	0.2908	0.6614	0.5123
Review Agents												
AgentReview	✓			✓	○	0.5160	0.5074	0.5528	0.0042	0.0026	0.6411	0.3707
AIScientist	✓			✓	○	0.7100	0.5417	0.6418	0.3162	0.2508	0.7493	0.5350
SEA-7B	✓			✓	○	0.6960	0.4104	0.5459	0.0983	0.0754	0.6585	0.4141
CycleReviewer-7B	✓			✓	○	0.6620	0.5300	0.6766	0.3132	0.2255	0.6859	0.5155
DeepReview-7B	✓		△	✓	○	0.6540	0.5470	0.5890	0.2928	0.2110	0.6938	0.4979
DeepReview-14B	✓		△	✓	○	0.6860	0.6240	0.6494	0.3995	0.2991	0.6657	0.5540
Comparative Systems												
PairReview		✓		✓	○	0.6700	0.5701	0.6047	0.2585	0.1781	0.6644	0.4910
CNPE-7B		✓		✓	○	0.7200	0.6692	0.7363	0.3995	0.2774	<u>0.8040</u>	0.6010
NAIP-8B			✓		○	0.6140	0.5413	0.5641	0.1585	0.1074	0.6882	0.4456
NAIPv2-8B		✓			○	<u>0.7260</u>	<u>0.6780</u>	<u>0.7627</u>	<u>0.4205</u>	0.2928	0.7723	<u>0.6087</u>
Ours												
GraphReview	✓	✓	✓	✓	○	0.8980	0.8806	0.9590	0.6626	0.4808	0.8547	0.7893

Table 1: Comparison of method characteristics and performance. ✓ indicates the use of a corresponding information source or the availability of a functionality; △ denotes a feature that has been developed but is not enabled by default in the released evaluation setting; and ●/○ indicate high/low risks of data leakage. **Best** and second-best results are highlighted, respectively.

Variant	Avg. Decision	Avg. Ranking	Avg. Performance
Information Sources			
w/o Graph	0.7640	0.5836	0.6738
w/o Intrinsic Quality	0.8978	0.6127	0.7553
w/o Synchronic Links	0.8918	<u>0.6657</u>	<u>0.7787</u>
w/o Diachronic Links	0.8979	0.6339	0.7659
Training Strategies			
w/o SFT	<u>0.8606</u>	<u>0.6067</u>	<u>0.7337</u>
w/o RIML	0.8604	0.5903	0.7254
Graph Construction			
Random Connection	<u>0.8731</u>	<u>0.6389</u>	<u>0.7560</u>
Aggregation Alternatives			
Naïve Win-Rate Averaging	0.8980	0.6461	0.7721
Bradley-Terry	<u>0.9030</u>	<u>0.6506</u>	<u>0.7768</u>
Borda Count	0.8585	0.6357	0.7471
Full Model			
GraphReview	0.9125	0.6660	0.7893

Table 2: Ablation study results of different information sources and training strategies. The full model always achieves the **best** results, and the second-best results in each group are underlined.

Ablation Study

We conduct comprehensive ablation studies to assess the effects of information fusion, training strategies, graph construction, and aggregation methods. The results are summarized in Table 2.

We first assess the contribution of the three information sources. Excluding any source from graph construction reduces average performance. In particular, the full model outperforms the baseline that removes the graph structure and relies solely on point priors by 17.1%, confirming the critical role of graph-based modeling. Moreover, removing Intrinsic Quality, Diachronic Links, or Synchronic Links causes rela-

tive decreases of 4.3%, 3.0%, and 1.3%, respectively. These results demonstrate that multi-source fusion effectively captures complementary signals of paper quality.

We next examine the training strategies. Removing the SFT stage reduces average performance by 7.0%, while removing RIML leads to an 8.1% decrease. This suggests that SFT introduces valuable review knowledge, whereas the soft-target objective of RIML further improves model alignment. The benefits of the two stages are complementary.

We then investigate the graph construction strategy. Under the same experimental setting, S2FM achieves 4.4% higher average performance than random connectivity, indicating that semantically selected neighbors provide a more informative graph structure.

Finally, we compare PPR with alternative aggregation methods. PPR outperforms naïve win-rate averaging, Bradley-Terry modeling, and Borda count method. Overall, the full GraphReview model consistently achieves the best performance, confirming the effectiveness of its designs.

Review Text Quality

We compare the review quality of GraphReview against baselines. We randomly sample 200 papers and adopt an LLM-as-a-Judge protocol to evaluate the aggregated reviews, with response order randomized to mitigate positional bias. GPT-5.4 serves as the evaluator, while all evaluated models belong to different model families from the evaluator to reduce potential bias. As shown in Table 3, each pairwise comparison assesses technical depth, evidence grounding, scientific rigor, revision utility, and overall preference.

The within-family evaluation shows that GraphReview substantially outperforms naïve LLM-based review generation using the same backbone, achieving overall preference

vs Baseline	Technical Depth	Evidence Ground- ing	Scientific Rigor	Revision Utility	Overall Preference
GraphReview (Gemini-2.5-Flash) vs Naïve LLM (Gemini)					
Gemini-2.5-Flash	80.0/19.5	63.5/35.0	89.0/9.5	93.5/6.5	88.5/11.5
Gemini-3-Flash	70.0/30.0	58.0/41.0	82.0/17.5	91.0/8.0	82.0/18.0
GraphReview (DeepSeek-V3.2) vs Naïve LLM (DeepSeek)					
DeepSeek-V3.2	76.0/24.0	63.0/35.0	83.5/14.0	95.5/3.0	84.5/15.5
DeepSeek-V4-Pro	86.0/13.0	78.0/21.0	85.5/13.5	95.0/4.5	85.5/14.0
GraphReview (DeepSeek-V3.2) vs Other Baselines					
DeepReview-7B	92.5/7.5	95.0/5.0	97.0/3.0	99.0/1.0	97.5/2.5
DeepReview-14B	63.5/36.5	66.5/33.0	76.5/23.5	63.0/35.0	69.0/31.0
CNPE-7B	91.0/8.0	97.0/2.5	97.0/2.5	99.0/0.5	97.0/2.5

Table 3: Win/Loss rates (%; Ties are omitted) of our review texts with different consolidation model against same-family naïve LLMs, and other baselines.

win rates of 88.5% with Gemini-2.5-Flash and 84.5% with DeepSeek-V3.2. Replacing the naïve baselines with stronger next-generation models has only a limited impact on the win rates (Gemini: -6.5; DeepSeek: +1.0), suggesting that the improvements mainly arise from graph-based structured information fusion rather than increased backbone quality.

GraphReview also outperforms other baselines across all evaluation dimensions, achieving overall preference win rates above 50%.

Hyperparameter Analysis

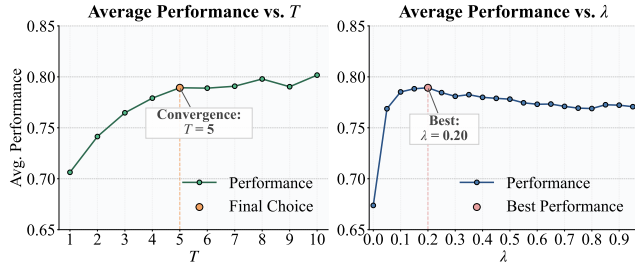


Figure 3: Hyperparameter analysis, including the convergence of model performance as T increases (left) and the selection of the optimal damping factor λ (right).

We further analyze the key hyperparameters in Figure 3. We first examine how overall performance changes as T increases. Here, T controls inference-time expansion by activating more S2FM semantic edges and therefore increasing the amount of structured graph evidence available for aggregation. The results exhibit a clear convergence pattern: performance improves steadily in the first few iterations, while the gains become marginal after $T > 5$. This observation indicates that $T = 5$ achieves a favorable balance between effectiveness and efficiency.

We then examine the effect of λ . Setting $\lambda = 0$ corresponds to relying solely on point priors. Incorporating graph information consistently yields strong performance, with limited sensitivity to λ . The best result is achieved at $\lambda = 0.20$. These results validate the effectiveness of graph-based evidence aggregation and show that moderate relational evidence can substantially refine the initial estimates while remaining robust across different values of λ .

Generalization

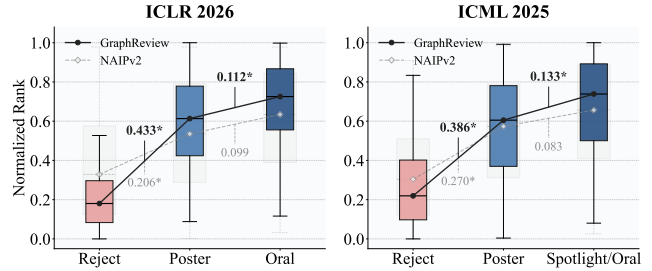


Figure 4: Generalization results. The y-axis shows normalized rank from 0 to 1, where higher values indicate higher predicted paper quality. The annotated values denote the median difference between groups, and * indicates statistical significance with $p < 0.001$.

We evaluate generalization in two settings. The cross-time-period setting is based on ICLR 2026 and includes three categories: Rejected, Poster, and Oral. The cross-venue setting is based on ICML 2025 and includes Rejected, Poster, and Spotlight/Oral. For each conference, we randomly sample 500 papers while keeping the class distribution as balanced as possible. As percentile ranks do not follow a normal distribution, we use the nonparametric Mann-Whitney U test and compare GraphReview with the strongest baseline, NAIPv2-8B, under the same setting.

The results show significant differences across all categories, with a consistent trend across both datasets. The gap between rejected and poster papers is substantial, reaching 0.433 on ICLR 2026 and 0.386 on ICML 2025. In contrast, the gap among accepted papers is smaller, at 0.112 and 0.133, respectively. This pattern is consistent with review practice: accepted and rejected papers usually differ clearly in quality, whereas quality differences among accepted papers are more subtle. These results indicate that our method captures both coarse-grained and fine-grained distinctions in paper quality.

Compared with NAIPv2-8B, GraphReview increases the Reject-vs-Poster gap by 0.227 on ICLR 2026 and by 0.116 on ICML 2025. It also increases the accepted-paper gap by 0.013 and 0.050 on the two datasets, respectively. Moreover, unlike GraphReview, NAIPv2-8B does not yield a significant Poster-vs-Spotlight/Oral gap ($p > 0.001$). These results suggest that GraphReview generalizes more effectively across time periods and venues when identifying differences in paper quality.

Conclusion

We propose GraphReview, an LLM-based graph framework for scientific paper evaluation that integrates complementary evidence through inference-time graph evidence expansion. Experiments show that it achieves leading performance on quality-ranking, decision-prediction, and review-generation benchmarks, while demonstrating strong generalization. Overall, GraphReview provides a data-driven and holistic framework for leveraging graph-structured review evidence, representing a meaningful step toward automated scientific paper evaluation.

References

- Alberts, B.; Hanson, B.; and Kelner, K. L. 2008. Reviewing peer review.
- Chang, Y.; Li, Z.; Zhang, H.; Kong, Y.; Wu, Y.; So, H. K.-H.; Guo, Z.; Zhu, L.; and Wong, N. 2025. TreeReview: A dynamic tree of questions framework for deep and efficient LLM-based scientific peer review. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 15662–15693.
- Dong, J.; An, S.; Yu, Y.; Zhang, Q.-W.; Luo, L.; Huang, X.; Wu, Y.; Yin, D.; and Sun, X. 2025. Youtu-graphrag: Vertically unified agents for graph retrieval-augmented complex reasoning. *arXiv preprint arXiv:2508.19855*.
- Du, J.; Wang, Y.; Wang, W.; Deng, Z.; Liu, S.; Lou, R.; Zou, H. P.; Venkit, P. N.; Zhang, N.; Srinath, M.; et al. 2024. LLMs assist NLP researchers: Critique paper (meta-) reviewing. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, 5081–5099.
- Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Metropolitansky, D.; Ness, R. O.; and Larson, J. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.
- Feng, T.; Sun, Y.; and You, J. 2025. GraphEval: A Lightweight Graph-Based LLM Framework for Idea Evaluation. In Yue, Y.; Garg, A.; Peng, N.; Sha, F.; and Yu, R., eds., *International Conference on Learning Representations*, volume 2025, 99181–99209.
- Finkelshtein, B.; Cucerzan, S.; Jauhar, S. K.; and White, R. 2025. Actions Speak Louder than Prompts: A Large-Scale Study of LLMs for Graph Inference. *arXiv preprint arXiv:2509.18487*.
- Fortunato, S.; Bergstrom, C. T.; Börner, K.; Evans, J. A.; Helbing, D.; Milojević, S.; Petersen, A. M.; Radicchi, F.; Sinatra, R.; Uzzi, B.; et al. 2018. Science of science. *Science*, 359(6379): eaao0185.
- Frasca, F.; Bar-Shalom, G.; Ziser, Y.; and Maron, H. 2025. Neural Message-Passing on Attention Graphs for Hallucination Detection. *arXiv preprint arXiv:2509.24770*.
- Garg, M. K.; Prasad, T.; Singhal, T.; Kirtani, C.; Mandal, M.; and Kumar, D. 2025. ReviewEval: An Evaluation Framework for AI-Generated Reviews. *arXiv preprint arXiv:2502.11736*.
- Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, 1263–1272. Pmlr.
- Guo, Z.; Xia, L.; Yu, Y.; Ao, T.; and Huang, C. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2(3).
- Gutiérrez, B. J.; Shu, Y.; Gu, Y.; Yasunaga, M.; and Su, Y. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37: 59532–59569.
- Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.
- He, G.; Sun, A.; and Lu, W. 2023. Research explosion: More effort to climb onto shoulders of the giant. *arXiv preprint arXiv:2307.06506*.
- He, G.; Xue, Z.; Jiang, Z.; Kang, Y.; Zhao, S.; and Lu, W. 2023. H2CGL: Modeling dynamics of citation network for impact prediction. *Information Processing & Management*, 60(6): 103512.
- Höppner, N.; Eshuijs, L.; Alivanistos, D.; Zamprogno, G.; and Tiddi, I. 2025. Automatic Evaluation Metrics for Artificially Generated Scientific Research. *arXiv preprint arXiv:2503.05712*.
- Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.
- Jeh, G.; and Widom, J. 2003. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, 271–279.
- Jin, Y.; Zhao, Q.; Wang, Y.; Chen, H.; Zhu, K.; Xiao, Y.; and Wang, J. 2024. AgentReview: Exploring Peer Review Dynamics with LLM Agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 1208–1226.
- Jing, Z.; Zeng, Q.; Fang, R.; Sun, Y.; Wang, B.; and Hu, P. 2026. Entropy-Guided Dynamic Tokens for Graph-LLM Alignment in Molecular Understanding. *arXiv preprint arXiv:2602.02742*.
- Joshi, C. K. 2025. Transformers are graph neural networks. *arXiv preprint arXiv:2506.22084*.
- Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Latona, G. R.; Ribeiro, M. H.; Davidson, T. R.; Veselovsky, V.; and West, R. 2024. The ai review lottery: Widespread ai-assisted peer reviews boost paper scores and acceptance rates. *arXiv preprint arXiv:2405.02150*.
- Li, C.; Hu, X.; Xu, M.; Li, K.; Zhang, Y.; and Cheng, X. 2025. Can Large Language Models Be Trusted Paper Reviewers? A Feasibility Study. *arXiv preprint arXiv:2506.17311*.
- Lu, C.; Lu, C.; Lange, R. T.; Foerster, J.; Clune, J.; and Ha, D. 2024. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.
- Lu, K.; Xu, S.; Li, J.; Ding, K.; and Meng, G. 2025. Agent Reviewers: Domain-specific Multimodal Agents with Shared Memory for Paper Review. In *Forty-second International Conference on Machine Learning*, 40803–40830.
- Luo, L.; Zhao, Z.; Liu, J.; Qiu, Z.; Dong, J.; Panev, S.; Gong, C.; Vu, T.-T.; Haffari, G.; Phung, D.; et al. 2025. G-reasoner: Foundation Models for Unified Reasoning over Graph-structured Knowledge. *arXiv preprint arXiv:2509.24276*.
- Qwen Team. 2024. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*.
- Tan, C.; Lyu, D.; Li, S.; Gao, Z.; Wei, J.; Ma, S.; Liu, Z.; and Li, S. Z. 2024. Peer review as a multi-turn and long-context dialogue with role-based interactions. *arXiv preprint arXiv:2406.05688*.

- Tang, J.; Yang, Y.; Wei, W.; Shi, L.; Su, L.; Cheng, S.; Yin, D.; and Huang, C. 2024. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 491–500.
- Tao, H.; Zhang, Y.; Tang, Z.; Peng, H.; Zhu, X.; Liu, B.; Yang, Y.; Zhang, Z.; Xu, Z.; Zhang, H.; et al. 2025. Code graph model (cgm): A graph-integrated large language model for repository-level software engineering tasks. *arXiv preprint arXiv:2505.16901*.
- Thakkar, N.; Yuksekgonul, M.; Silberg, J.; Garg, A.; Peng, N.; Sha, F.; Yu, R.; Vondrick, C.; and Zou, J. 2025. Can llm feedback enhance review quality? a randomized study of 20k reviews at iclr 2025. *arXiv preprint arXiv:2504.09737*.
- Thakkar, N.; Yuksekgonul, M.; Silberg, J.; Garg, A.; Peng, N.; Sha, F.; Yu, R.; Vondrick, C.; and Zou, J. 2026. A large-scale randomized study of large language model feedback in peer review. *Nature Machine Intelligence*, 1–11.
- Tyser, K.; Segev, B.; Longhitano, G.; Zhang, X.-Y.; Meeks, Z.; Lee, J.; Garg, U.; Belsten, N.; Shporer, A.; Udell, M.; et al. 2024. Ai-driven review systems: evaluating llms in scalable and bias-aware academic reviews. *arXiv preprint arXiv:2408.10365*.
- Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.
- Wang, D.; Zuo, Y.; Lu, G.; and Wu, J. 2025. UniGTE: Unified Graph-Text Encoding for Zero-Shot Generalization across Graph Tasks and Domains. *arXiv preprint arXiv:2510.16885*.
- Wang, H.; Feng, S.; He, T.; Tan, Z.; Han, X.; and Tsvetkov, Y. 2023. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems*, 36: 30840–30861.
- Weng, Y.; Zhu, M.; Bao, G.; Zhang, H.; Wang, J.; Zhang, Y.; and Yang, L. 2025. CycleResearcher: Improving Automated Research via Automated Review. In *The Thirteenth International Conference on Learning Representations*.
- Wu, L.; Wang, D.; and Evans, J. A. 2019. Large teams develop and small teams disrupt science and technology. *Nature*, 566(7744): 378–382.
- Xue, Z.; He, G.; Jiang, Z.; Gu, S.; Kang, Y.; Zhao, S.; and Lu, W. 2024. Predicting scientific impact through diffusion, conformity, and contribution disentanglement. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2764–2774.
- Yu, J.; Ding, Z.; Tan, J.; Luo, K.; Weng, Z.; Gong, C.; Zeng, L.; Cui, R.; Han, C.; Sun, Q.; et al. 2024. Automated Peer Reviewing in Paper SEA: Standardization, Evaluation, and Analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 10164–10184.
- Zeng, S.; Tian, K.; Zhang, K.; Wang, Y.; Gao, J.; Liu, R.; Yang, S.; Li, J.; Long, X.; Ma, J.; et al. 2025. ReviewRL: Towards Automated Scientific Review with RL. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 16942–16954.
- Zhang, J. 2023. Graph-toolformer: To empower llms with graph reasoning ability via prompt augmented by chatgpt. *arXiv preprint arXiv:2304.11116*.
- Zhang, Y.; ZHANG, H.; Ji, W.; Hua, T.; Haber, N.; Cao, H.; and Liang, W. 2025. From Replication to Redesign: Exploring Pairwise Comparisons for LLM-Based Peer Review. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Zhao, J.; Zhuo, L.; Shen, Y.; Qu, M.; Liu, K.; Bronstein, M.; Zhu, Z.; and Tang, J. 2023. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*.
- Zhao, P.; Tian, J.; Xing, Q.; Zhang, X.; Li, Z.; Qian, J.; Cheng, M.-M.; and Li, X. 2025a. NAIPv2: Debiased Pairwise Learning for Efficient Paper Quality Estimation. *arXiv preprint arXiv:2509.25179*.
- Zhao, P.; Xing, Q.; Dou, K.; Tian, J.; Tai, Y.; Yang, J.; Cheng, M.-M.; and Li, X. 2025b. From Words to Worth: Newborn Article Impact Prediction with LLM. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 1183–1191.
- Zheng, P.; Yao, J.; Zheng, J.; Gu, C.; He, G.; Liu, J.; Huang, Y.; Guo, T.; and Lu, W. 2026a. From Isolated Scoring to Collaborative Ranking: A Comparison-Native Framework for LLM-Based Paper Evaluation. *arXiv preprint arXiv:2603.17588*.
- Zheng, W.; Xu, Y.; Lin, X.; Gao, C.; Wang, W.; and Feng, F. 2026b. Navigating Through Paper Flood: Advancing LLM-Based Paper Evaluation Through Domain-Aware Retrieval and Latent Reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 35041–35049.
- Zhou, R.; Chen, L.; and Yu, K. 2024. Is LLM a reliable reviewer? a comprehensive evaluation of LLM on automatic paper reviewing tasks. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024)*, 9340–9351.
- Zhu, M.; Weng, Y.; Yang, L.; and Zhang, Y. 2025. DeepReview: Improving LLM-based Paper Review with Human-like Deep Thinking Process. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 29330–29355.
- Zhuang, Z.; Chen, J.; Xu, H.; Jiang, Y.; and Lin, J. 2025. Large language models for automated scholarly paper review: A survey. *Information Fusion*, 103332.

Use of Generative AI

Generative artificial intelligence was used in this study only for language support, including grammar correction, phrasing refinement, and style standardization. It was not involved in the paper’s conceptualization, scientific analysis, or substantive writing. All AI-assisted content was reviewed, verified, and, where necessary, revised by the authors in accordance with ethical standards. The authors take full responsibility for the paper’s content and conclusions.

Theoretical Implications

A growing line of work in automated paper evaluation seeks to mimic the human review process. We take a different view. In our opinion, *the more closely a method reproduces the procedure by which humans review papers, the further it may be from an ideal decision process*. Human reviewing involves rich internal reasoning and highly personal judgments that are not reflected in the final written review. A strong review system should therefore learn from the evidence-based reasoning underlying human judgment, rather than from the surface form of review texts.

Therefore, AI-based paper evaluation does not need to follow the same paradigm as human reviewing. Machines offer distinct advantages. Unlike human reviewers, they can draw on much larger volumes of data and perform fine-grained comparisons between a paper and a wide range of relevant evidence. This differs fundamentally from human practice. Due to limited time, human reviewers often cannot make such extensive and detailed empirical connections, and instead rely on broad impressions formed through prior experience in the field. This is exactly why we introduce graph structure into the review process: to make the evidence chain explicit while exploiting the large-scale processing capabilities of machines. At the same time, humans retain unique strengths, particularly intuition for innovation and for identifying important problems. Such intuition is grounded in embodied experience and creativity. Current AI systems do not yet possess this ability, and closing this gap should remain an important direction for future research.

In the long run, AI-assisted reviewing should move beyond imitating human review and instead develop into a review paradigm with its own strengths: comprehensive, data-intensive, and systems-oriented. Such a paradigm can complement human capabilities more effectively in a future of human-AI collaboration. From this perspective, our work represents a meaningful step toward that goal.

Full Literature Review

Automated Paper Review

The most common line of work evaluates a paper’s inherent attributes, using LLM-driven agent review frameworks to simulate and reproduce the peer-review process with automated pointwise scoring (Jin et al. 2024; Lu et al. 2024), often augmented by iterative, multi-round conversational reviewing (Weng et al. 2025; Tan et al. 2024). Many approaches further improve performance through training, transforming open-source models into expert reviewers via domain-specific fine-tuning or reinforcement learning (Zeng et al.

2025). For example, Yu et al. (2024) fine-tunes models to generate customized feedback aligned with human preferences (Tyser et al. 2024; Garg et al. 2025). Other methods employ agents to construct tree-structured workflows to enhance generation quality (Chang et al. 2025). More recent systems combine model training with agent techniques, such as integrating evidence-based reasoning and structured analysis into multi-stage workflows (Zhu et al. 2025), or building multi-agent systems with prompt engineering, shared memory, and multimodal perception (Lu et al. 2025).

However, recent studies suggest that pointwise scoring is sensitive to inconsistencies in rating scales across venues and time. As a result, many works shift to parallel submissions within the same venue and develop comparison-based scoring strategies. Examples include predicting paper quality scores via pairwise comparisons (Zhao et al. 2025a), performing pairwise comparisons and aggregating preferences to derive a global ranking (Zhang et al. 2025; Höpner et al. 2025; Zheng et al. 2026b), and combining similarity-based sampling with comparison-reward-driven reinforcement learning (Zheng et al. 2026a).

Despite incorporating broader domain information, most methods still overlook recent, non-contemporaneous knowledge in the field. Such knowledge reflects patterns of scholarly communication and inheritance and is crucial for assessing the true contribution of a new paper. Although integrating non-contemporaneous knowledge has been explored in graph-based impact prediction methods (He et al. 2023; Xue et al. 2024), it remains underexplored in LLM-based systems. While Zhu et al. (2025) includes a retrieval module in its workflow, it does not empirically validate its effectiveness. Only Zhao et al. (2025b) makes substantial use of this information by retrieving related papers via external tools and predicting standardized impact scores through listwise ranking.

Overall, existing approaches almost always study these information sources in isolation and rarely integrate them effectively. A promising and important direction is to consider them jointly within a unified graph-based system.

Graph Methods for LLMs

Before the rise of large language models, the dominant paradigm for graph tasks was message-passing-based graph neural networks (GNNs), which perform representation learning and downstream prediction via neighborhood aggregation (Kipf and Welling 2016; Veličković et al. 2017; Gilmer et al. 2017; Hamilton, Ying, and Leskovec 2017).

As LLMs have advanced in language understanding and knowledge representation, graph methods have gradually become complementary to LLMs. Some early approaches rewrote local graph structures into natural language descriptions using handcrafted rules and fed them into models as prompts (Wang et al. 2023; Zhao et al. 2023).

Recent studies aim to align graph structure and linguistic semantics at the levels of representation space and reasoning mechanisms. Through instruction tuning or adapter modules, they endow LLMs with structured graph understanding. For example, GraphGPT (Tang et al. 2024) integrates LLMs with graph structural knowledge via graph instruc-

tion tuning; UniGTE (Wang et al. 2025) unifies structure and semantics by attending over graph tokens and natural language, enabling cross-task reasoning; Jing et al. (2026) achieves efficient alignment between a frozen graph encoder and a frozen LLM; G-reasoner (Luo et al. 2025) unifies graph and language foundation models into a general graph representation for scalable reasoning; Tao et al. (2025) incorporates repository code graphs into attention and maps node attributes through adapters; and Feng, Sun, and You (2025) use a lightweight graph-based LLM framework for idea evaluation.

Meanwhile, another line of work leverages graphs for retrieval-augmented generation, using knowledge graphs as external knowledge bases. Representative examples include GraphRAG (Edge et al. 2024), which introduces a community summarization and stepwise aggregation framework; LightRAG (Guo et al. 2024), which combines graph structure with vector representations via a two-level retrieval scheme; HippoRAG (Gutiérrez et al. 2024), which performs memory-style retrieval with personalized PageRank; and an integrated system that adopts agent-based ideas to unify retrieval, graph construction, and community detection (Dong et al. 2025).

Other approaches treat graphs as interactive environments, mitigating LLM limitations in multi-step logical reasoning, precise computation, and spatiotemporal awareness through multi-step planning and tool use (Zhang 2023), including generating code to execute graph algorithms (Finkelshtein et al. 2025).

However, for semantically centered paper review problems, graph methods that primarily rely on modality alignment or external retrieval are not directly reusable. The texts in our setting exhibit fine-grained semantic and argumentative relations that are tightly coupled with the review task. Forcing text into node embeddings, or making results explicit via alignment and averaging, can lead to semantic compression and partially undermine the transparency and interpretability of review texts. We therefore turn to a redesign of the overall graph-based approach.

Method Details

Sequential 2-Factor Matching

We propose a Sequential 2-Factor Matching algorithm to efficiently construct a sparse comparison adjacency matrix with time complexity $O(TN)$, where $T \ll N$. The full procedure is given in Algorithm 2. Let $\mathbf{C}^{(t)} \in \mathbb{R}^{N \times N}$ denote the matching state after expansion step t , where $c_{uv}^{(t)}$ indicates whether an edge has been activated between nodes u and v . At each step, we solve a maximum-weight 2-factor matching problem to obtain $\Delta \mathbf{C}^{(t)}$, which adds high-similarity edges based on pairwise inner products $\mathbf{h}_u^\top \mathbf{h}_v$. The constraints enforce binary and symmetric edge assignments, require each node to connect to exactly two new neighbors in the current step, and exclude previously selected edges. We then update the graph by $\mathbf{C}^{(t)} = \mathbf{C}^{(t-1)} + \Delta \mathbf{C}^{(t)}$ and repeat the process for the fixed budget T . Thus, T controls comparison coverage, and the final degree of each node is $2T$.

This matching procedure has several desirable properties:

Algorithm 2: Sequential 2-Factor Matching for Evidence Expansion

Require: Node embeddings $\{\mathbf{h}_u\}_{u=1}^N$, graph-expansion budget T
Ensure: Adjacency matrix $\mathbf{C}^{(T)}$

- 1: **Initialize** $t \leftarrow 0$, $\mathbf{C}^{(0)} \leftarrow \mathbf{0}$
- 2: **while** $t < T$ **do**
- 3: Obtain $\Delta \mathbf{C}^{(t)}$ by solving:

$$\begin{aligned} \max_{\Delta c_{uv}^{(t)}} \quad & \sum_{u=1}^n \sum_{v=1}^n \mathbf{h}_u^\top \mathbf{h}_v \cdot \Delta c_{uv}^{(t)} \\ \text{s.t.} \quad & \sum_{v=1}^n \Delta c_{uv}^{(t)} = 2, \\ & \Delta c_{uv}^{(t)} = \Delta c_{vu}^{(t)}, \\ & \Delta c_{uv}^{(t)} \in \{0, 1\}, \\ & \Delta c_{uv}^{(t)} \leq 1 - c_{uv}^{(t-1)}. \end{aligned}$$

- 4: $\mathbf{C}^{(t)} \leftarrow \mathbf{C}^{(t-1)} + \Delta \mathbf{C}^{(t)}$
 - 5: $t \leftarrow t + 1$
 - 6: **end while**
 - 7: **return** $\mathbf{C}^{(T)}$
-

1. $\mathbf{C}^{(T)}$ is the sum of T disjoint 2-factors and satisfies greedy subset inclusion: $\mathbf{C}^{(1)} \subset \mathbf{C}^{(2)} \subset \dots \subset \mathbf{C}^{(T)}$, which ensures consistent behavior under scaling. The final degree of each node is $D = 2T$.
2. Assuming a unique optimal 2-factor exists, the algorithm is permutation equivariant, meaning the optimal result is invariant to node relabeling.
3. For $T \geq 1$ and $N \geq 3$, the loop executes at least once, so the graph always contains edges. This follows from the fact that any complete graph K_N with $N \geq 3$ admits a 2-factor.

Text Consolidation

In practice, for node-level inference, an LLM’s output can be decomposed into two types of information, denoted as $f_{\text{LLM}}(x_v, p_s) = (\hat{y}_s, \hat{t}_s)$. Here, $\hat{y}_s$ is a multi-class probabilistic label distribution aligned with the training objective, which can be further converted into a scalar score, and $\hat{t}_s$ is a natural-language rationale for the review. Similarly, for edge-level inference, the model output is $f_{\text{LLM}}(x_u, x_v, p_c) = (\hat{y}_c, \hat{t}_c)$, where $\hat{y}_c$ represents a preference signal such as $x_u \succ x_v$ and is used directly for score construction, while $\hat{t}_c$ provides the corresponding natural-language rationale.

PPR operates on numerical priors and directed preference transitions rather than textual rationales. We therefore introduce a parallel text-processing procedure alongside PPR-based ranking aggregation. Specifically, PPR numerically aggregates the pointwise prior and directed preferences over the fixed graph, whereas the rationale texts associated with nodes and edges are collected, organized, and merged in parallel.

Following common review-writing conventions, we further reformat these scattered text fragments into a coherent evaluation paragraph. We design a lightweight prompt that instructs a model to reformat the collected texts. Under this prompt, the model consistently restructures information

from multiple sources and produces a consolidated evaluation with improved organization and coherence. We implement this step via an external instruction-following model as a text refinement module, instantiated as DeepSeek-V3.2 in our experiments. This module is model-agnostic and can be replaced by any instruction-following LLM with basic text organization capabilities.

The refined evaluation text serves two purposes. First, it presents the semantic content in a clearer and more readable form for human authors and reviewers. Second, it enables qualitative comparison with texts generated by other methods in our experiments.

Iterative Prompt Evolving

Paper reviewing is a specialized and cognitively demanding task. Even domain experts may find it challenging to articulate evaluation criteria that are both explicit and sufficiently fine-grained. To obtain prompts that are robust and comprehensive enough to cover the multiple dimensions of paper assessment, we develop a task-specific evolving procedure for prompt construction. As shown in Figure 5 (right), this process can be written as $p^{(0)} \rightarrow p^{(1)} \rightarrow \dots \rightarrow p^{(M)}$. Starting from an initial prompt $p^{(0)}$, the teacher model iteratively refines the prompt over M rounds. At each step, the model compares the outputs produced before and after refinement, and retains the prompt that leads to the better result. The final prompt, denoted by $p^* = p^{(M)}$, is then used to generate cold-start responses. These responses further serve as SFT data that teach the student model the evaluation dimensions and criteria underlying paper review decisions.

The update at iteration m is defined as:

$$p^{(m)} = f_{\text{Judger}}(p^{(m-1)}, f_{\text{Evolver}}(p^{(m-1)})) \quad (7)$$

At iteration m , the procedure takes the current best prompt from the previous round, $p^{(m-1)}$, as input and produces a refined candidate through f_{Evolver} . The function f_{Judger} then compares the current prompt with its refined version and selects the one that yields the better outcome. If the refined prompt improves performance, it is accepted as new $p^{(m)}$; otherwise, the previous prompt is preserved. Both f_{Evolver} and f_{Judger} are implemented with a designated teacher model through prompt-based optimization. In our implementation, we use DeepSeek-V3.2 as the teacher model by default.

Reward-Induced Maximum Likelihood

RIML is used in training for both node and edge capabilities. At the node level, the target distribution is softly induced by distance-based rewards over score anchors. At the edge level, the target distribution is a binary one-hot label induced by the pairwise ranking relation weighted by the score difference. For the complete training pipeline, see Figure 5 (right).

Node-level training For each sample v , let s_v denote its ground-truth score and let the input query be $q_v = (x_v, p_s)$, where x_v is the paper content and p_s is the prompt for the single-paper scoring task.

Because review scores in many top conferences are not restricted to integer values, we introduce a set of score anchors

to approximate the continuous score space using a finite set of proxy classes:

$$\mathbf{a} = \{a_1, a_2, \dots, a_K\} \quad (8)$$

The model outputs logits over these proxy class tokens:

$$\mathbf{z}_v = (z_{v1}, z_{v2}, \dots, z_{vK}) \in \mathbb{R}^K \quad (9)$$

The predictive distribution is defined as:

$$P(k | q_v) = \frac{\exp(z_{vk})}{\sum_{l=1}^K \exp(z_{vl})} \quad (10)$$

Rather than hard-assigning s_v to its nearest anchor, we define a distance-based reward for each anchor a_k :

$$r_{vk} = -\frac{(s_v - a_k)^2}{2\sigma^2} \quad (11)$$

Here, σ controls the smoothness of the supervision distribution. The reward-induced target distribution is then given by:

$$y_{vk} = \frac{\exp(r_{vk})}{\sum_{l=1}^K \exp(r_{vl})} \quad (12)$$

The node-level objective is defined as:

$$\mathcal{L}_{\text{node}}(\theta) = -\frac{1}{N} \sum_{v=1}^N \sum_{k=1}^K w_v y_{vk} \log P(k | q_v) \quad (13)$$

We set $w_v = 1$ for all samples. This soft supervision preserves relative positional information in the continuous score space through anchor-based distance modeling.

Edge-level training For each sample pair (u, v) , let s_u and s_v denote their ground-truth scores. We require $|s_u - s_v| > \delta$ to ensure sufficiently reliable supervision. We further construct training pairs with a greedy matching strategy such that each sample appears in at most one pair per training epoch, which improves pair diversity. The input query is $q_{uv} = (x_u, x_v, p_c)$, where x_u and x_v are the contents of the two papers and p_c is the prompt for the pairwise comparison task.

We formulate pairwise comparison as a binary classification problem. The label indicating whether sample u receives a higher score than sample v is:

$$k_{uv} = \mathbb{I}[s_u - s_v > 0] \quad (14)$$

The model outputs logits over two proxy class tokens:

$$\mathbf{z}_{uv} = (z_{uv0}, z_{uv1}) \in \mathbb{R}^2 \quad (15)$$

The predictive distribution is defined as:

$$P(k | q_{uv}) = \frac{\exp(z_{uvk})}{\exp(z_{uv0}) + \exp(z_{uv1})} \quad (16)$$

Unlike node-level training, edge-level training uses a one-hot target distribution induced by the ground-truth comparison relation:

$$r_{uvk} = \mathbb{I}[k = k_{uv}], \quad y_{uvk} = r_{uvk} \quad (17)$$

The edge-level objective is defined as:

$$\mathcal{L}_{\text{edge}}(\theta) = -\frac{1}{M} \sum_{m=1}^M \sum_{k=0}^1 w_{uv} y_{uvk} \log P(k | q_{uv}) \quad (18)$$

Here, M is the upper bound on the total number of valid edges, and each pair (u, v) corresponds to one edge index m . We define the edge weight as $w_{uv} = |s_u - s_v|$. A larger score gap indicates a clearer ranking relation and thus provides a more reliable supervision signal.

Experiment Details

Dataset Construction

As shown in Figure 5 (left), we first collect the full text of all papers through the OpenReview API, parse the PDFs with MinerU, an open-source OCR tool, and convert them into Markdown text. We then use the pre-trained semantic encoder Qwen3-8B-Embedding to generate paper embeddings and store them in a local vector database for subsequent graph construction.

We construct the primary supervised dataset using the full set of ICLR 2025 submissions, following the same split protocol as DeepReview. Specifically, the average human reviewer score serves as the ground-truth label for paper quality. The ICLR 2025 data are split into training, validation, and test sets, with approximately 8K papers for training and validation combined and 0.5K papers for testing. The training and validation sets are further divided at a ratio of 9:1. To build the retrieval-augmented external knowledge base, we additionally collect all accepted papers from ICLR, ICML, and NeurIPS in 2023 and 2024, yielding a total of about 15K papers. This auxiliary corpus is designed to capture recent research trends and provide up-to-date background knowledge for retrieval and graph construction.

To further evaluate cross-year transferability and generalization, we additionally sample 500 papers each from ICLR 2026 and ICML 2025 for out-of-domain evaluation, while using the same external knowledge base. These evaluation sets include expert-provided paper group annotations and cover multiple quality levels.

Training Setup and Hyperparameters

The model is trained in two stages. In the SFT stage, the learning rate is fixed to 2×10^{-4} , the batch size is 1, and gradient accumulation spans 4 steps. In the RIML stage, we use AdamW with learning rates of 1×10^{-4} , a batch size of 2, gradient accumulation over 16 steps, and a cosine learning rate scheduler. All training and inference experiments are conducted on a single RTX Pro 6000 96G GPU. The total training time is approximately 10 hours. During inference, we use vLLM for text generation. During inference-time graph evidence expansion, we set n , the total number of test papers, to 500. Since incorporating the entire external knowledge base would incur substantial computational cost, we set N to 1000 for efficiency and randomly sample these instances across different years to validate the effectiveness of the model. The minimum improvement threshold ϵ is set to

0.01, and the maximum patience c_{max} is set to 3. The damping factor λ in PPR is consistently set to 0.2, in accordance with the best result from the hyperparameter study. For the anchors vector $\mathbf{a}$ in node-level training, we adopt the ICLR 2025 rating scale levels $\{1, 3, 5, 6, 8, 10\}$. For the score difference threshold δ in edge-level training, the value is set to 1.5. All random seeds set to 42.

Baselines

We reproduce a diverse set of baselines to benchmark our method from multiple perspectives. To ensure a fair evaluation, we explicitly account for potential data leakage issues, enabling a systematic comparison with both traditional and contemporary approaches.

Naive LLMs For this family of models, we directly query general-purpose LLMs via the APIs for score prediction. All samples use the same scoring prompt template, which is obtained through our proposed prompt evolution mechanism, consistent with the prompt template implemented in our scoring model. During inference, the temperature is fixed at 0, and all other parameters remain at their default values. We do not use tool calling, external retrieval, or any other auxiliary components. The model produces the final prediction in a single inference round.

Classic GNNs We implement a set of GNN baselines on the paper graph, where each node represents a submission and node features are derived from the paper text. For each paper, we concatenate the title and abstract, and encode the resulting text with Qwen3-Embedding-8B.

We build a semantic kNN graph using cosine distance in the embedding space. For each node, we connect it to its top- k nearest neighbors according to cosine distance computed from the pretrained text embeddings. In all settings, we remove duplicate edges when needed and add self-loops to every node. We set $k = 10$. We formulate the task as node-level regression and train all GNNs to predict the scalar target score using MSE. All experiments use a unified DGL implementation for graph construction.

We evaluate three standard two-layer architectures: GCN, GAT, and GraphSAGE. The GCN baseline contains two graph convolution layers, with ReLU activation and dropout applied after the first layer. The GAT baseline uses an 8-head attention layer followed by a single-head output layer. The first layer uses ELU activation, and both attention dropout and feature dropout follow standard practice. The GraphSAGE baseline adopts the mean aggregator in both layers, with ReLU and dropout applied between layers. The hidden dimension is set to 128 for all models. For GAT, this dimension is evenly split across attention heads. We optimize all models with Adam using a learning rate of 1×10^{-3} , train for at most 200 epochs, and apply early stopping based on validation loss with a patience of 5 epochs. The dropout rate is 0.5 for GCN and GraphSAGE, and 0.6 for GAT, reflecting the common observation that attention-based GNNs often benefit from stronger regularization. All hyperparameters are selected through careful multi-round tuning.

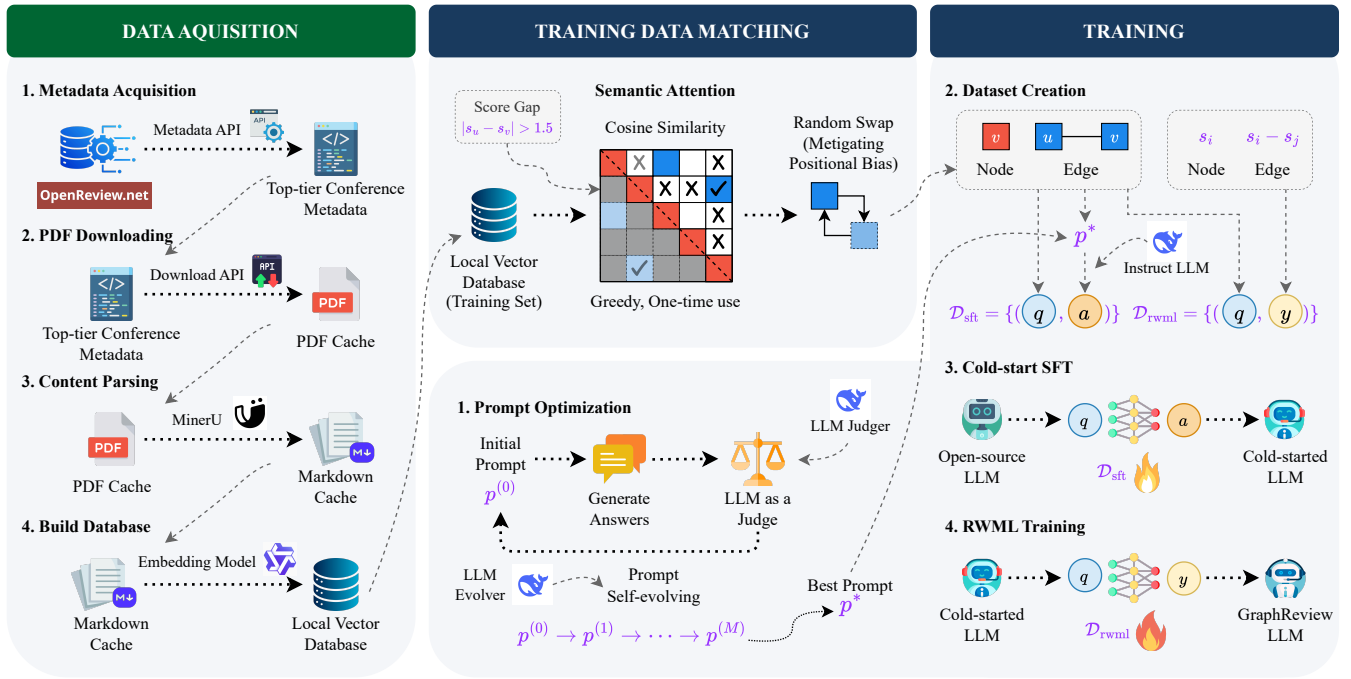


Figure 5: Pipeline of dataset construction (left) and training process (right).

Review Agents and Comparative Systems We compare multiple categories of LLM-based reviewing systems, including general agentic reviewing frameworks, specialized reviewing models released by their authors, and comparative methods that require retraining under our data split. To ensure fair comparison and reliable conclusions, we standardize the backbone model, input information, and data split whenever possible, and assess potential data leakage on a case-by-case basis.

For AIScientist, AgentReview, and PairReview, we use GPT-oss-120B as the unified backbone model. In our experiments, these systems take the full paper as input and output the final score prediction. For SEA-7B, CycleReviewer-7B, and NAIPv1-8B, we directly evaluate the open-source models released by their authors. For DeepReview-7B, DeepReview-14B, and CNPE-7B, the original authors trained these models on the ICLR 2025 dataset, and their train-test split is broadly aligned with ours. There is no overlap between the test samples used in this paper and the training data of these models. Therefore, evaluating them on our test set does not introduce data leakage. For NAIPv2-8B, the original ICLR 2025 split used in its reported experiments differs from ours. Consequently, directly using the released weights would make it difficult to ensure a fair comparison under a unified evaluation protocol. We therefore retrain this model on our split using the training code released by the authors, and report results on the same test set.

Evaluation Metrics

We evaluate model performance from two complementary perspectives: **binary decision quality** and **ranking quality**. The former assesses whether the predicted decision matches

the ground-truth label, while the latter evaluates the quality of the predicted ordering, which is particularly important for downstream applications such as recommendation, retrieval, and literature intelligence.

For binary decision evaluation, we report **Accuracy**, **F1**, and **AUC**. For ranking evaluation, we report **Spearman’s ρ** , **Kendall’s τ** , and **NDCG@10**.

Accuracy Accuracy is defined as:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}[\hat{d}_i = d_i] \quad (19)$$

Here, $d_i \in \{0, 1\}$ and $\hat{d}_i \in \{0, 1\}$ denote the ground-truth and predicted binary labels, respectively.

F1 F1, namely Macro-F1, is computed by first calculating the F1 score for each class $c \in \mathcal{C}$:

$$F1_c = \frac{2 \text{Precision}_c \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \quad (20)$$

And then averaging across classes:

$$F1 = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} F1_c \quad (21)$$

AUC AUC measures the probability that a randomly selected positive instance is assigned a higher predicted score than a randomly selected negative instance:

$$\text{AUC} = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \left(\mathbb{I}[\hat{y}_i > \hat{y}_j] + \frac{1}{2} \mathbb{I}[\hat{y}_i = \hat{y}_j] \right) \quad (22)$$

Name	Decision Performance			Ranking Performance			Avg. Performance
	Accuracy	F1	AUC	Spearman ρ	Kendall τ	NDCG@10	
Gemini-3-Flash-Preview	0.6780	0.6230	0.7278	0.5128	0.4181	0.8136	0.6289
GLM-5	0.6540	0.5949	0.6737	0.4018	0.3274	0.8144	0.5777
Qwen3.6-Plus	0.6660	0.6089	0.7174	0.4575	0.3708	0.6390	0.5766
DeepSeek-V4-Pro	0.6780	0.6230	0.7146	0.4046	0.3213	0.6829	0.5707
GraphReview	0.8980	0.8806	0.9590	0.6626	0.4808	0.8547	0.7893

Table 4: Performance of recently released LLMs. **Best** and second-best results are highlighted.

Here, $\mathcal{P}$ and $\mathcal{N}$ denote the sets of positive and negative instances.

Spearman’s ρ Spearman’s ρ measures the correlation between the ground-truth and predicted rankings:

$$\rho = 1 - \frac{6 \sum_{i=1}^N \Delta_i^2}{N(N^2 - 1)} \quad (23)$$

Here, Δ_i denotes the difference between the rank of y_i and that of $\hat{y}_i$.

Kendall’s τ Kendall’s τ measures pairwise rank agreement with tie correction:

$$\tau = \frac{n_c - n_d}{\sqrt{(n_0 - n_x)(n_0 - n_y)}} \quad (24)$$

Here, n_c and n_d denote the numbers of concordant and discordant pairs, $n_0 = N(N - 1)/2$, and n_x and n_y denote the numbers of pairs tied only in the predicted and ground-truth rankings, respectively.

NDCG@10 NDCG@10 evaluates the quality of the top-ranked results:

$$\text{NDCG@10} = \frac{\text{DCG@10}}{\text{IDCG@10}} \quad (25)$$

Specifically:

$$\text{DCG@10} = \sum_{i=1}^{10} \frac{2^{y_{(i)}} - 1}{\log_2(i + 1)} \quad (26)$$

$y_{(i)}$ is the ground-truth relevance of the item ranked at position i , and IDCG@10 is the maximum achievable DCG@10.

Additional Results

Performance of Recent LLMs

In addition to the main LLM baselines, we also evaluate several recently released models. Their results are reported in Table 4. Since all of these models were released after December 2025, they carry substantial leakage risk with respect to our benchmark. We therefore include them only as a reference for empirical comparison. Despite this favorable condition, their performance remains consistently below that of GraphReview. This result further indicates that stronger general-purpose LLMs alone do not close the gap in paper review, while our framework retains a clear advantage.

Bootstrap Confidence Intervals

We report 95% bootstrap confidence intervals for the main experimental results in Table 5. The intervals are computed on the same test set as the main experimental table. GraphReview maintains a clear advantage over all baselines in decision, ranking, and average performance. The overall performance interval of GraphReview, is separated from that of the strongest baseline NAIPv2-8B, indicating that the improvement is robust under bootstrap resampling.

Full Ablation Results

For completeness, we report the full ablation results on all evaluation metrics in Table 6. Although several alternatives are competitive on individual ranking metrics, the full model achieves the best overall performance, confirming that graph construction, graph propagation, and model training jointly contribute to the final system.

Difference Tests of Generalization

We also provide the detailed statistical test results for the generalization analysis in Table 7. Since rankings do not naturally follow a normal distribution, non-parametric tests are used. For both conferences, the Kruskal-Wallis test shows significant overall differences across decision categories. The pairwise Mann-Whitney U tests are also significant for all category pairs. The result pattern aligns with practical review intuition: accepted and rejected papers usually differ substantially in quality, whereas accepted papers have already reached a relatively high standard, making within-group differences less pronounced. These results provide additional evidence that the predicted ranking scores align well with real review outcomes and remain discriminative across different decision levels.

Proof

Theorem 1. *For every integer $N \geq 3$, the complete graph K_N contains a 2-factor.*

Proof. Let $V(K_N) = \{v_0, v_1, \dots, v_{N-1}\}$. Consider the cycle $C = (v_0, v_1, v_2, \dots, v_{N-1}, v_0)$. Since K_N is complete, it contains every edge of the form $\{v_i, v_{i+1 \bmod N}\}$ for $i \in \{0, 1, \dots, N-1\}$, so C is a well-defined subgraph of K_N . Moreover, C is spanning because it contains every node of K_N . For each node v_i , the two incident edges in C are $\{v_i, v_{i-1 \bmod N}\}$ and $\{v_i, v_{i+1 \bmod N}\}$, and these two edges are distinct since $N \geq 3$. Hence every node has degree exactly 2 in C . Therefore, C is a 2-factor of K_N . $\square$

Method	Decision Performance			Ranking Performance			
	Accuracy	F1	AUC	Spearman ρ	Kendall τ	NDCG@10	Avg. Performance
Classic GNNs							
GCN	0.5980 [0.5560,0.6380]	0.5404 [0.4943,0.5807]	0.5720 [0.5194,0.6203]	0.1267 [0.0352,0.2060]	0.0852 [0.0231,0.1398]	0.5552 [0.4767,0.6441]	0.4129 [0.3508,0.4715]
GAT	0.5760 [0.5340,0.6200]	0.5235 [0.4798,0.5684]	0.5104 [0.4562,0.5627]	0.0485 [-0.0395,0.1359]	0.0349 [-0.0255,0.0953]	0.5734 [0.4595,0.6728]	0.3778 [0.3108,0.4425]
GraphSAGE	0.5640 [0.5240,0.6080]	0.5143 [0.4697,0.5600]	0.5055 [0.4522,0.5588]	0.0492 [-0.0364,0.1420]	0.0348 [-0.0239,0.1004]	0.5868 [0.4739,0.7175]	0.3758 [0.3099,0.4478]
Naïve LLMs							
GPT-5-Mini	0.7100 [0.6740,0.7520]	0.6605 [0.6186,0.7055]	0.7104 [0.6661,0.7537]	0.4136 [0.3406,0.4889]	0.3317 [0.2724,0.3954]	0.6982 [0.5969,0.8109]	0.5874 [0.5281,0.6511]
Gemini-2.5-Flash	0.6060 [0.5620,0.6500]	0.5387 [0.4942,0.5832]	0.5557 [0.5194,0.5933]	0.1936 [0.0995,0.2724]	0.1569 [0.0813,0.2190]	0.6422 [0.5468,0.7368]	0.4488 [0.3838,0.5091]
DeepSeek-V3.2	0.6180 [0.5760,0.6600]	0.5527 [0.5070,0.5967]	0.5975 [0.5749,0.6212]	0.3536 [0.2817,0.4242]	0.2908 [0.2319,0.3496]	0.6614 [0.5407,0.7618]	0.5123 [0.4520,0.5689]
Review Agents							
AgentReview	0.5160 [0.4760,0.5601]	0.5074 [0.4666,0.5511]	0.5528 [0.5031,0.6050]	0.0042 [-0.0759,0.0889]	0.0026 [-0.0591,0.0673]	0.6411 [0.5133,0.7722]	0.3707 [0.3040,0.4408]
AIScientist	0.7100 [0.6700,0.7500]	0.5417 [0.4950,0.5890]	0.6418 [0.5936,0.6885]	0.3162 [0.2416,0.3973]	0.2508 [0.1919,0.3168]	0.7493 [0.5853,0.8507]	0.5350 [0.4629,0.5987]
SEA-7B	0.6960 [0.6560,0.7360]	0.4104 [0.3961,0.4240]	0.5459 [0.4934,0.5951]	0.0983 [0.0108,0.1878]	0.0754 [0.0083,0.1449]	0.6585 [0.5317,0.7196]	0.4141 [0.3494,0.4679]
CycleReviewer-7B	0.6620 [0.6200,0.7040]	0.5300 [0.4835,0.5748]	0.6766 [0.6244,0.7259]	0.3132 [0.2362,0.3855]	0.2255 [0.1696,0.2808]	0.6859 [0.5856,0.7795]	0.5155 [0.4532,0.5751]
DeepReview-7B	0.6540 [0.6100,0.6960]	0.5470 [0.5010,0.5899]	0.5890 [0.5342,0.6413]	0.2928 [0.2052,0.3736]	0.2110 [0.1477,0.2719]	0.6938 [0.6360,0.7880]	0.4979 [0.4390,0.5601]
DeepReview-14B	0.6860 [0.6440,0.7240]	0.6240 [0.5816,0.6657]	0.6494 [0.6009,0.6959]	0.3995 [0.3172,0.4772]	0.2991 [0.2362,0.3610]	0.6657 [0.5903,0.7578]	0.5540 [0.4950,0.6136]
Comparative Systems							
PairReview	0.6700 [0.6300,0.7120]	0.5701 [0.5267,0.6208]	0.6047 [0.5526,0.6587]	0.2585 [0.1750,0.3428]	0.1781 [0.1201,0.2383]	0.6644 [0.5756,0.7442]	0.4910 [0.4300,0.5528]
CNPE-7B	0.7200 [0.6800,0.7580]	0.6692 [0.6229,0.7091]	0.7363 [0.6855,0.7825]	0.3995 [0.3172,0.4763]	0.2774 [0.2197,0.3344]	0.8040 [0.7133,0.8991]	0.6010 [0.5398,0.6599]
NAIP-8B	0.6140 [0.5740,0.6560]	0.5413 [0.4961,0.5842]	0.5641 [0.5146,0.6189]	0.1585 [0.0757,0.2469]	0.1074 [0.0498,0.1700]	0.6882 [0.5896,0.7812]	0.4456 [0.3833,0.5095]
NAIPv2-8B	0.7260 [0.6880,0.7640]	0.6780 [0.6336,0.7211]	0.7627 [0.7144,0.8044]	0.4205 [0.3443,0.4941]	0.2928 [0.2392,0.3480]	0.7723 [0.6975,0.8720]	0.6087 [0.5528,0.6673]
Ours							
GraphReview	0.8980 [0.8720,0.9240]*	0.8806 [0.8514,0.9108]*	0.9590 [0.9424,0.9747]*	0.6626 [0.6092,0.7108]*	0.4808 [0.4351,0.5237]*	0.8547 [0.7756,0.9397]	0.7893 [0.7476,0.8306]*

Table 5: Main results with 95% bootstrap confidence intervals. * denotes that GraphReview is significantly better than the second-best method under a paired bootstrap test with $p < 0.001$. **Best** and **second-best** point estimates are highlighted.

Graph Evidence Expansion as Paradigm

Graph-based LLM paper review can be developed into a complete and independent evidence-aggregation paradigm. GraphReview is not only a standalone method, but also a unified abstraction for a broader class of approaches that organize pointwise and pairwise review signals in a common graph structure. This abstraction is distinct from representation-learning GNNs: the graph is used as a symbolic structure for inference-time evidence expansion and global ranking aggregation. To support this view, we revisit mainstream methods through the lens of optimization objectives and show that their core ideas can be naturally unified within a graph-based framework. In particular, most existing LLM-based paper review methods can be characterized by two classic objectives.

Pointwise Objective The first line of work emphasizes accurate scoring of individual papers and adopts a pointwise objective:

$$\phi_s^* = \arg \min_{\phi_s} \sum_{x_v \in \mathbf{x}} \mathcal{L}(f_s(x_v; \phi_s), s_v) \quad (27)$$

Here, f_s denotes a scoring function for estimating the quality of a single paper, with the objective of aligning the predicted score with the ground-truth label.

Pairwise Objective The second line of work focuses on comparison and discrimination and adopts a pairwise objective:

$$\phi_c^* = \arg \min_{\phi_c} \sum_{x_u, x_v \in \mathbf{x}} \mathcal{L}(f_c(x_v, x_u; \phi_c), s_u, s_v) \quad (28)$$

Here, f_c denotes a comparison function over paper pairs, whose goal is to capture relative preference or comparative quality.

Combined Objective From a graph perspective, pointwise and pairwise estimation are not separate formulations, but complementary signals that can be organized and globally integrated within a common evidence graph. We therefore write the combined optimization objective as:

$$\Phi^* = \arg \min_{\Phi} \mathcal{L} \left((f_{\text{agg}} \circ ((f_c \circ f_{\text{att}}) \oplus f_s))(\mathbf{x}; \Phi), \mathbf{r}^{\text{gt}} \right) \quad (29)$$

Here, f_{att} defines the graph construction strategy by selecting informative paper pairs, f_c and f_s generate edge-level and node-level signals, respectively, and f_{agg} aggregates these signals over the graph to produce the final prediction aligned with the ground-truth ranking $\mathbf{r}^{\text{gt}}$.

	Decision Performance			Ranking Performance			
Variant	Accuracy	F1	AUC	Spearman ρ	Kendall τ	NDCG@10	Avg. Performance
Information Sources							
w/o Graph	0.7580	0.7167	0.8174	0.5529	0.3948	0.8032	0.6738
w/o Intrinsic Quality	0.8820	0.8618	0.9497	0.5903	0.4243	0.8236	0.7553
w/o Synchronic Links	0.8820	0.8618	0.9315	<u>0.6632</u>	0.4823	<u>0.8514</u>	<u>0.7787</u>
w/o Diachronic Links	0.8820	0.8618	0.9498	0.6541	0.4730	0.7745	0.7659
Training Strategies							
w/o RWML	0.8500	0.8244	0.9068	0.5857	0.4187	0.7665	0.7253
w/o SFT	0.8500	0.8244	0.9073	0.6036	0.4331	0.7835	0.7337
Graph Construction							
Random Connection	0.8620	0.8384	0.9189	0.6447	0.4642	0.8077	0.7560
Aggregation Alternatives							
Naïve Win-Rate Averaging	0.8820	0.8618	0.9503	0.6422	0.4619	0.8342	0.7721
Bradley-Terry	<u>0.8900</u>	<u>0.8712</u>	0.9478	0.6652	0.4785	0.8082	0.7768
Borda Count	0.8460	0.8197	0.9097	0.6298	0.4527	0.8246	0.7471
Full Model							
GraphReview	0.8980	0.8806	0.9590	0.6626	<u>0.4808</u>	0.8547	0.7893

Table 6: Full ablation results with detailed decision and ranking metrics. **Best** and second-best results in each metric are highlighted.

Conference	Kruskal-Wallis Test		Pairwise Mann-Whitney U Test					
	H-statistic	p-value	Reject vs Poster		Reject vs Spotlight/Oral		Poster vs Spotlight/Oral	
			U-statistic	p-value	U-statistic	p-value	U-statistic	p-value
ICLR 2026	273.0119	0.000*	2086.0	0.000*	962.0	0.000*	10460.0	0.000*
ICML 2025	157.2580	0.000*	7182.0	0.000*	1927.0	0.000*	9247.0	0.000*

Table 7: Results of Kruskal-Wallis and pairwise Mann-Whitney U tests across decision categories. * denotes statistical significance with $p < 0.001$.

Specifically, in GraphReview, f_{att} is instantiated as Sequential 2-Factor Matching, which selects informative comparison edges with low overhead. The aggregation function f_{agg} is instantiated as Personalized PageRank, which computes a stationary ranking over the fixed directed preference graph while incorporating the pointwise prior. The functions f_c and f_s correspond to the pairwise comparison model and the pointwise scoring model obtained from two-stage training.

Modular Decoupling A key property of this framework is modular decoupling. GraphReview does not depend on any fixed backend model; rather, it lies in offering a graph-based formulation that can uniformly integrate heterogeneous review signals. In principle, any mainstream LLM or existing paper review method can serve as a backend. Its pointwise or pairwise predictions can be embedded into the graph and then combined through graph aggregation to produce a unified global ranking.

Experiments To validate the effectiveness of this generalized graph evidence-expansion paradigm, we further replace the representative pair selection and comparison component $f_c \circ f_{\text{att}}$ and the direct scoring component f_s with several classic methods, so that they provide edge-level and node-level review signals, respectively. We keep Personalized PageRank as f_{agg} .

We conduct ablation studies over different module combinations. Specifically, we evaluate two representative settings: (1) CNPE-7B (Zheng et al. 2026a) combined with DeepReview-14B (Zhu et al. 2025), and (2) PairReview (Zhang et al. 2025) combined with CycleReviewer-7B (Weng

et al. 2025).

The results reported in Table 8 and 9 show that graph-based fusion consistently integrates heterogeneous review signals and outperforms each individual method on most metrics. For the combination of CNPE-7B and DeepReview-14B, the fused model achieves the best performance on all six metrics, yielding an average relative improvement of approximately 4.3% over the strongest standalone baseline, CNPE-7B. For the combination of PairReview and CycleReviewer-7B, the fused model also performs best on most metrics and improves the average performance by approximately 4.9% relative to the strongest single model, CycleReviewer-7B. By integrating diverse approaches within a graph evidence structure, this paradigm captures substantial complementary information and further improves overall performance.

Overall, these results suggest that graph-based LLM paper review could be viewed not only as a specific method, but also as a general inference-time evidence-expansion paradigm for paper evaluation that unifies the two dominant formulations of pointwise and pairwise estimation with strong scalability and compatibility.

Methodological Design Details

Beyond Classic GNNs

Classic GNNs (Kipf and Welling 2016; Veličković et al. 2017; Gilmer et al. 2017; Hamilton, Ying, and Leskovec 2017) treat text embeddings as node features and then learn node representations through iterative neighborhood aggregation. However, our experiments show that such models perform poorly on paper reviewing and fall substantially

Variant	Decision Performance			Ranking Performance			Avg. Performance
	Accuracy	F1	AUC	Spearman ρ	Kendall τ	NDCG@10	
CNPE-7B	0.7180	0.6698	0.7354	0.3950	0.2753	0.7752	0.5948
DeepReview-14B	0.6860	0.6240	0.6494	0.3995	0.2991	0.6657	0.5540
Combined	0.7300	0.6839	0.7388	0.4604	0.3250	0.7838	0.6203

Table 8: Performance of graph-based fusion with CNPE-7B and DeepReview-14B. **Best** and second-best results are highlighted.

Variant	Decision Performance			Ranking Performance			Avg. Performance
	Accuracy	F1	AUC	Spearman ρ	Kendall τ	NDCG@10	
PairReview	0.6380	0.5762	0.6080	0.2440	0.1670	0.6843	0.4863
CycleReviewer-7B	0.6620	0.5300	0.6766	0.3132	0.2255	0.6859	0.5155
Combined	0.6780	0.6230	0.6794	0.3450	0.2359	0.6826	0.5407

Table 9: Performance of graph-based fusion with PairReview and CycleReviewer-7B. **Best** and second-best results are highlighted.

behind GraphReview. The two paradigms both use graph structure, but GraphReview is not a representation-learning GNN: it elicits task-specific pointwise and pairwise evidence from original paper texts and aggregates the resulting fixed preference graph numerically. This distinction motivates the following analysis of why conventional GNN operators are unsuitable for the task.

We answer this question through a mechanistic analysis. Using GraphSAGE as a representative example, we argue that this family of methods incurs substantial information loss throughout the pipeline and largely undermines the structural clarity and interpretability of the graph. The central issue is that its generic embedding transmission, neighborhood pooling, and latent state mixing operators are not designed for evaluation tasks that require fine-grained semantic understanding and strong alignment with the downstream objective.

Generic Embedding Transmission When edges do not carry explicit semantic features or types, message passing is insensitive to the task-specific relations. Prior methods propagate information mainly according to graph topology. For example, GraphSAGE transmits embeddings with the same generic rule regardless of the specific relation between two papers. In practice, however, paper-paper relations are grounded in meaningful scholarly contexts. Papers from different periods may reflect inheritance or intellectual influence, while contemporaneous papers may represent direct competition. Simply passing information between neighboring nodes cannot capture these distinctions, because the propagation process discards the task-relevant context that gives each relation its meaning.

Neighborhood Pooling Existing aggregation functions compress neighborhood information through permutation-invariant operations, but this compression distorts the actual contribution of neighbors. GraphSAGE uses mean, sum, or max pooling to aggregate incoming messages. As a result, semantically unrelated or even contradictory signals are merged into a single representation, and differences in neighbor importance are flattened by averaging or pooling. This can dilute neighbor-specific evidence, especially under heterophily and suppress higher-order semantic cues, such as

argumentative stance and strength of evidence. Such behavior is fundamentally misaligned with our goal of identifying truly contributive papers.

Latent State Mixing The update function further mixes intrinsic textual information with semantically heterogeneous external signals, even though these two sources represent different forms of evidence. In paper reviewing, node information reflects the intrinsic quality of the manuscript, whereas edge information reflects the additional value introduced by its domain-level relations to other papers. Traditional GraphSAGE-style methods concatenate these signals into a single embedding and process them jointly in a black-box fashion. This design not only fails to explicitly preserve relation-specific evidence, which is crucial for reviewing, but also weakens attribution, which is critical in the reviewing process.

Classical GNNs can be effective for relatively shallow tasks such as node classification and link prediction, but they are ill-suited for paper reviewing. This task requires deep semantic understanding, rigorous reasoning, and careful assessment of novelty. Under such demands, conventional GNNs exhibit clear and fundamental limitations.

By addressing the limitations of traditional GNN architectures for paper reviewing, GraphReview establishes a graph-structured evidence framework tailored to this task. Instead of recursively transmitting latent text embeddings, it uses LLMs to elicit explicit comparative evidence on S2FM-selected pairs. It preserves neighbor-specific textual rationales for consolidation and separately converts pairwise outcomes into a directed preference graph. PPR then combines this graph with pointwise priors to obtain globally coupled and attributable predictions. These design choices collectively allow GraphReview to substantially outperform conventional GNN baselines.

Beyond LLM Agents

A natural question is: *why not build a conventional agent-based reviewer, but instead rely on a structured external graph?* Answering this question also clarifies the main challenge addressed in this paper.

At the core of this issue is the fact that paper review is not a closed task, but a process whose judgments become

clearer through interaction with an open scientific environment. Classical LLM agent architectures (Lu et al. 2024; Weng et al. 2025) typically treat paper review as a task to be solved through decomposition, iterative reflection, or retrieval-augmented analysis. Although retrieved evidence may provide useful background knowledge, it is usually incorporated only as auxiliary context (Zhu et al. 2025) rather than as a structured signal that directly shapes evaluation. As a result, these methods often fail to capture the factors that truly determine review quality, such as methodological novelty, the relationship between a paper and prior work, and the temporal lineage of ideas across the literature.

This limitation reflects a deeper mismatch between conventional agent architectures and the nature of the review task. General-purpose agents are designed for broad language tasks through iterative textual reasoning. In contrast, paper review requires reasoning over the external scientific context of a paper, including citation dependencies, conceptual genealogies, and cross-paper comparisons. Treating review as an isolated text understanding problem therefore ignores the critical role of these external signals.

Our method is based on a different perspective. We explicitly model the scientific context as a graph of review evidence and integrate local judgments through sparse pairwise comparison and PPR-based ranking aggregation. This design enables the system to interact with structured knowledge beyond the target paper and to combine review-relevant evidence in a task-aligned manner. We argue that it is precisely this capability, rather than more elaborate in-text reasoning alone, that allows our method to achieve more reliable and accurate paper evaluation.

Training Objectives

The training objective for the LLMs should ideally satisfy two requirements. First, it must capture review signals at both levels: the score distribution over nodes and the binary preference relation over edges. Second, it should retain the semantic evidence associated with expert judgment, namely the evidence supporting each judgment. Direct regression on review scores is insufficient for both purposes. It does not align well with the target distribution and may fail to retain the semantic evidence contained in expert reviews. To address this limitation, we adopt a two-stage training framework.

Paper evaluation is better characterized as a preference comparison problem rather than a long-horizon reasoning task such as mathematical deduction. In peer review, judgments are often formed by directly comparing how two papers present the same aspect, such as clarity, novelty, or empirical support, at corresponding textual locations. This process relies more on evidence localization and comparative attention over text than on repeated Markov state transitions or search over a reasoning tree. Therefore, RLVR methods such as GRPO are not a natural fit for this setting. This mismatch is both methodological and computational. RLVR-based approaches are typically expensive to train, and their reward signals are often sparse, which may lead to unstable optimization. Moreover, even with reward supervision, large language models can produce reasoning traces that appear plausible without being faithful. Sparse rewards may further

exacerbate this issue. By contrast, our RIML training strategy is more naturally aligned with preference optimization methods, whose objective is to match human comparative judgments rather than solve a multi-step reasoning problem.

Once the task is framed as preference alignment, the central question becomes what should be aligned. Supervising only the final decision may be insufficient, because the relevant signal does not reside solely in the conclusion. If training targets only the final outcome, the model may learn to imitate answer patterns without internalizing the underlying criteria. Therefore, we distill the global explanatory rationale behind a preference judgment into the earliest stage of generation. We first generate the explanation supporting the preference from the teacher model, and then use this distilled signal to fine-tune an open-source student model. After training, the model’s initial token can implicitly encode a compact representation of the subsequent explanatory reason, helping the model reach preference decisions without relying on lengthy explicit reasoning at inference time. Our ablation results show that directly applying preference optimization without this explanatory cold-start SFT causes the model to overlook critical evidence and yields worse performance.

Additionally, our approach is primarily intended for review scenarios where supervision is derived from human rating signals. Under the assumption that a paper’s derivations and experiments are broadly sound and follow standard academic practice, the model is particularly well suited to capturing human scientific taste and assessing academic value. Its current formulation focuses less on settings that require rigorous verification of formal derivations or detailed validation of experimental results, which we leave for future work.

Case Study

Table 10 presents a complete case study using a representative paper. It covers the full workflow from the original manuscript to the final review report, offering an end-to-end walkthrough of our evaluation pipeline.

Prompts

We release all prompt templates used in our system in Tables 11–15 to document instruction design across evaluation stages and support reproducibility and analysis.

Case Study

We take paper 68J0pJFCi3 as an example and illustrate the full process it undergoes throughout evaluation.

Original Text

```
# ON REPRESENTING CONVEX QUADRATICALLY CON-STRAINED QUADRATIC PROGRAMS VIA GRAPH  
↪ NEURAL NETWORKS
```

Anonymous authors Paper under double-blind review

```
# ABSTRACT
```

```
Convex quadratically constrained quadratic programs (QCQPs) involve finding a  
↪ solution ...
```

```
# 1 INTRODUCTION
```

```
Quadratic programs (QPs) are a ...
```

On the one hand, we directly score this central paper to obtain an initial estimate of its quality at the node level.

Focal Paper Scoring $\times 1$

Paper (68J0pJFCi3, 2025)

```
<|SCORING_TOKEN|> This paper, \"On Representing Convex Quadratically Constrained  
↪ Quadratic Programs via Graph Neural Networks,\" proposes a novel tripartite graph  
↪ representation for convex QCQPs and establishes theoretical foundations regarding  
↪ the expressiveness of message passing GNNs applied to this representation ...
```

On the other hand, we iteratively introduce additional edges and compare the paper with other papers.

Synchronic and Diachronic Pairs Comparison $\times 2T$

Paper A (68J0pJFCi3, 2025), Paper B (ULleq1Dtaw, 2024)

```
<|COMPARISON_TOKEN|> The choice of Paper A as the higher quality paper is based on a  
↪ comprehensive analysis across six key criteria. While both papers present novel  
↪ contributions to video-language modeling, Paper A demonstrates superior ...
```

Paper A (XnVttczoAV, 2025), Paper B (68J0pJFCi3, 2025)

...

We convert these special tokens into numerical values and integrate the corresponding signals via the PPR algorithm, producing a final numerical score. Meanwhile, the textual content is processed in parallel through a consolidation step.

Final Report

****Ranking:**** 225/500

****Decision:**** Reject

****Summary**:**

This paper introduces a tripartite graph representation for convex Quadratically
↪ Constrained Quadratic Programs (QCQPs) and presents a theoretical framework
↪ linking this representation to the expressiveness of message passing Graph Neural
↪ Networks (GNNs). While the paper ...

****Advantages**:**

1. ****Relevant Problem Selection:**** The task of representing and solving convex QCQPs
↪ ...

****Disadvantages**:**

1. ****Incremental and Poorly Justified Technical Contribution:**** The proposed ...

****Questions**:**

1. The theorem states that a GNN can universally approximate the ...

****Suggestions**:**

1. ****Complete and Justify the Theoretical Claims:**** The authors ...

Finally, we produce a complete evaluation report with an associated score for paper 68J0pJFCi3.

Table 10: Case study. An example illustrating the complete workflow for evaluating a paper.

Criteria Optimization Prompt

```
You are an expert prompt optimizer.
Your task is to optimize the {criteria} so that the language model can generate better
↪ responses.
Do not provide any information related to the output format or output requirements;
↪ analyze only the content.
The <Criteria> you provide must be structured (Use 1. 2. 3. ...), expressed clearly and
↪ accurately.
You must only return the optimized <Criteria>.
Prompt:
```
{prompt}
```
Current <Criteria> (Empty if none):
```
{criteria}
```
```

Table 11: Criteria optimization prompt. Used to create a new prompt based on an existing prompt.

Answer Evaluation Prompt

```
You are an expert answer evaluator.
Your task is to compare which of the two answers is of higher quality.
You must only return the character (A/B) representing the quality.
Answer A:
```
{answer_A}
```
Answer B:
```
{answer_B}
```
Better:
```

Table 12: Answer evaluation prompt. Used to evaluate the quality of answers generated by evolved criteria.

Data Construction Prompt (Scoring)

```
You are an expert reviewer.
Evaluating a paper by its quality.
Your analysis must follow the following criteria:
{criteria}
Your answer should be about 2000 words.
Do not use bold or any other symbols.
Your answer must always begin with a scored number, and there must be no other text
↪ before it.
Do not mention any issues regarding the truncation of submitted content; truncation does
↪ not constitute part of the analysis.
The correct score for this paper is {ground_truth}.
You must give this score and provide a convincing explanation for why this score is
↪ appropriate.

Evaluate the following paper and provide a score using the following scale:
0: strong reject
1: reject, not good enough
2: marginally below the acceptance threshold
3: marginally above the acceptance threshold
4: accept, good paper
5: strong accept, should be highlighted at the conference
Here is the paper:
```
{paper_text}
```

Please provide your score first as a single number (0-5), then explain your reasoning.
Score:
```

Table 13: Data construction prompt for paper scoring.

Data Construction Prompt (Comparison)

You are an expert reviewer.
Analyze the following two papers separately, then indicate which one is of higher
↪ quality.
Your analysis must follow the following criteria:
{criteria}
Your answer should be about 1000 words.
Do not use bold or any other symbols.
Your answer must always begin with a choice (A or B), and there must be no other text
↪ before it.
Do not mention any issues regarding the truncation of submitted content; truncation does
↪ not constitute part of the analysis.
The correct choice is {ground_truth}.
You must output this choice and provide a convincing explanation for why this choice is
↪ appropriate.

Compare the following two papers and decide which one is better in quality.
Paper A:
...
{paper_text_a}
...
Paper B:
...
{paper_text_b}
...

Please provide your choice first as a single letter (A or B), then explain your
↪ reasoning.
Choice:

Table 14: Data construction prompt for pair comparison.

Text Consolidation Prompt

You are an expert academic reviewer and research analyst.
Your task is to produce an enhanced review of a single paper by integrating valuable
→ comparative insights.

Instructions:

1. Use the provided `single\_paper\_review` as the primary foundation and preserve its core
→ judgment unless the comparative evidence clearly justifies adjustment.
2. For each entry in `related\_pairs`, briefly extract only the most relevant information
→ from `pair\_comparison`, especially comparative strengths, weaknesses, missing
→ validations, or clearer methodological standards that are directly useful for
→ evaluating this paper.
3. Integrate these insights naturally into the `single\_paper\_review`, citing the relevant
→ literature in the merged text. Citation format: e.g. `( #0, 2025)`. Use comparisons
→ selectively and only when they strengthen or clarify the review.
4. You must output content related to `ranking` and `decision` at first, e.g.
→ `\*\*Ranking:\*\* (0/500)` and `\*\*Decision:\*\* Accept`. Make sure the ranking, decision,
→ and all arguments are fully consistent with each other after revision.
5. Structure the review clearly into layered sections: first give an overall assessment,
→ then list the most important strengths, then the most important weaknesses, and
→ finally concrete questions/suggestions. Avoid repetition across sections.
6. The questions and suggestions proposed must all be highly practical, specific,
→ feasible, and directly actionable for the authors to address.
7. Keep the tone professional, evidence-based, and concise. Avoid exaggerated claims or
→ unsupported criticism.
8. Only output the merged text. Do not include any other content.

Here is all the content related to the paper:

```
...  
{json_str}  
...
```

The output format you need to follow:

```
...  
**Ranking:**  
**Decision:**  
**Summary:**  
**Advantages:**  
**Disadvantages:**  
**Questions:**  
**Suggestions:**  
...
```

Table 15: Text consolidation prompt. Merge the texts to generate complete review.

Text Evaluation Prompt

You are a senior meta-reviewer evaluating the quality of peer-review reports for
→ scientific conferences and journals.

Your task is to compare two reviews of the same paper and decide which review is stronger
→ as a scientific review.

RESPONSE FORMAT REQUIREMENTS:

1. Return a valid JSON object only, with no extra text.
2. The JSON must contain exactly these ten keys:
 - "technical_depth"
 - "technical_depth_reason"
 - "evidence_grounding"
 - "evidence_grounding_reason"
 - "scientific_rigor"
 - "scientific_rigor_reason"
 - "revision_utility"
 - "revision_utility_reason"
 - "overall_preference"
 - "overall_preference_reason"
3. For each label key, the value must be exactly one of: "A", "B", or "Tie".
4. For each reason key, the value must be one brief sentence of at most 22 words.
5. Do not output anything except the JSON object.

EVALUATION DIMENSIONS:

1. technical_depth: Which review engages more deeply with the paper's technical
→ substance, such as method details, assumptions, derivations, proofs, experiments,
→ evaluation design, complexity, or implementation?
2. evidence_grounding: Which review ties its judgments more directly to paper-specific
→ evidence, claims, equations, tables, figures, baselines, metrics, or clearly missing
→ analyses?
3. scientific_rigor: Which review more rigorously evaluates validity, claim-evidence
→ alignment, fairness of comparisons, reproducibility, completeness of argumentation,
→ and whether the paper's conclusions are actually supported?
4. revision_utility: Which review gives more useful and actionable guidance for improving
→ the paper, especially through concrete, acceptance-relevant revisions?
5. overall_preference: Overall, which review is more valuable for editorial
→ decision-making and author revision, considering technical insight, evidence-based
→ criticism, exposure of substantive weaknesses, and usefulness for improving the
→ paper?

CORE JUDGING PRINCIPLES:

1. Judge only the quality of the reviews, not the quality of the paper.
2. Prefer reviews that identify central technical weaknesses, unsupported claims, weak
→ evidence, missing controls, incomplete proofs, confounds, unfair baselines, or
→ reproducibility gaps.
3. Prefer paper-specific critique over generic balance, polished wording, soft tone, or
→ formulaic reviewing language.
4. Do not reward a review merely for sounding more diplomatic, more moderate, more
→ balanced, or more polished.
5. Do not penalize a review merely for being critical, forceful, technically dense, or
→ highly detailed, if its concerns are concrete and grounded in the paper.
6. Strong reviews often directly explain why the current evidence is insufficient for the
→ paper's claims.
7. Comparative references to related work may be useful when they concretely support
→ criticism about novelty, baselines, theory, or evaluation standards; do not dismiss
→ them automatically unless they substantially replace paper-specific analysis.
8. Ignore superficial differences in politeness or rhetorical style unless they
→ materially affect scientific clarity or introduce unsupported claims.
9. If one review is sharper but better exposes acceptance-relevant weaknesses, it can be
→ better overall even if it is less smooth stylistically.
10. In close cases, prefer the review that better identifies substantive risks to
→ validity or acceptance.
11. If the two reviews are difficult to distinguish in quality, choose "Tie" rather than
→ defaulting to "A" due to positional bias.
12. If one review is empty, select the other review accordingly.

Continued on next page.

Table 16: Text evaluation prompt. Comparing the text quality with other approaches.

Text Evaluation Prompt (Continued)

Compare Review A and Review B as peer-review reports for the same paper.
Return only a valid JSON object with exactly these keys:

```
"technical_depth"  
"technical_depth_reason"  
"evidence_grounding"  
"evidence_grounding_reason"  
"scientific_rigor"  
"scientific_rigor_reason"  
"revision_utility"  
"revision_utility_reason"  
"overall_preference"  
"overall_preference_reason"
```

For each label key, output exactly one of: "A", "B", or "Tie".

For each reason key, output one brief sentence of at most 22 words.

Important instructions:

1. Judge only the quality of the reviews, not the paper itself.
2. Prefer reviews that identify important technical flaws, unsupported claims, weak
↪ evidence, missing experiments, incomplete proofs, unfair comparisons, or
↪ reproducibility issues.
3. Prefer paper-specific, evidence-linked criticism over smoother wording or more
↪ diplomatically balanced tone.
4. Do not reward a review merely for sounding more polished, more measured, or more
↪ conventionally editorial.
5. A sharper or more critical review can be better if its concerns are concrete,
↪ technically meaningful, and grounded in the paper.
6. Related-work comparisons may be useful when they concretely support criticism about
↪ novelty, baselines, theory, or evaluation standards.
7. In close cases, overall_preference should favor the review that better exposes
↪ acceptance-relevant weaknesses and better helps an editor decide.
8. If the two reviews are difficult to distinguish in quality, output "Tie" rather than
↪ defaulting to "A" because of positional bias.

Review A:

```
{review_a}
```

Review B:

```
{review_b}
```

Table 16: Text evaluation prompt (Continued). Comparing the text quality with other approaches.