EGTR-Review: Efficient Evidence-Grounded Scientific Peer Review Generation via Multi-Agent Teacher Distillation

Anonymous ACL submission

Abstract

Scientific peer review generation has attracted increasing attention for reducing reviewing burdens and providing timely feedback. However, existing Large Language Model (LLM)-based methods often produce generic comments with insufficient evidence support and weak source traceability, while complex multi-agent systems incur high inference costs. To address these challenges, we propose EGTR-Review, an Evidence-Grounded and Traceable Review Generation framework via Multi-Agent Teacher Distillation. EGTR-Review first constructs a multi-agent teacher that performs structure-aware paper decomposition, key-element extraction, external scholarly evidence retrieval, evidence-state labeling, verification reasoning, and review synthesis. It then distills both intermediate reasoning trajectories and final review comments into a lightweight student model through task-prefix-driven multi-task learning. An evidence-weighted objective further reduces the influence of weak, missing, or non-verifiable supervision. Experiments on public peer-review datasets show that EGTR-Review (Student) outperforms strong prompt-based, fine-tuned, and structured/agentic baselines across automatic metrics, LLM-as-Judge evaluation, and human evaluation, while maintaining strong factual grounding and source traceability with substantially lower token consumption and inference time. Our code, prompts, configurations, and sample data are available on GitHub [1].

1 Introduction

Scientific peer review is essential for research quality, but the rapid growth of submissions has increased reviewing burdens and delayed feedback (Kousha and Thelwall, 2024). Automated scientific peer review generation has therefore become an important direction in AI for science (D'Arcy et al., 2024). Recent LLM-based methods—prompt-based, supervised fine-tuning, and multi-agent—improve the fluency and structure of generated reviews (Robertson, 2023; Shamsabadi and D'Souza, 2024; Zhuang et al., 2025), but still struggle in real peer-review scenarios, especially in evidence support, source traceability, paper-specific feedback, and inference efficiency.

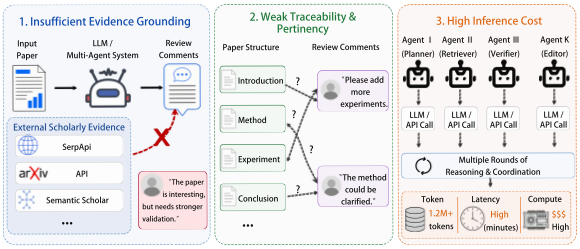

Figure 1: Key Limitations of Existing Multi-Agent Scientific Peer Review Generation Methods.

As summarized in Figure 1, existing methods face three main limitations. First, comments often lack support from paper content or external scholarly evidence, yielding generic rather than evidence-grounded feedback (Xu et al., 2026). Second, long and complex papers make it hard to trace comments to specific sections, paragraphs, or experimental settings, weakening traceability and pertinency (Li et al., 2024; Liu et al., 2024; Zhou et al., 2024). Third, multi-agent systems improve quality but rely on multi-round reasoning and repeated large-model calls, incurring high inference cost (D'Arcy et al., 2024). This raises a central question: how can evidence-grounded, traceable, and paper-specific review capabilities be transferred from a multi-agent teacher to a lightweight student?

In this work, we propose EGTR-Review, which builds a multi-agent teacher whose evidence-enhanced supervision integrates paper locations, scholarly evidence, evidence-state labels, reasoning trajectories, and final reviews; this is then dis-

[1] https://anonymous.4open.science/r/EGTR-Review-3BA8

tilled into EGTR-Review (Student). Unlike output-only distillation, the student learns the teacher's evidence-grounded reasoning process and final reviews through task-prefix-driven multi-task learning, while an evidence-weighted objective down-weights weak, missing, or non-verifiable supervision. At inference, it replaces costly teacher-side verification and synthesis but keeps paper parsing and evidence retrieval for traceability. Experiments on public peer-review datasets show that the Student outperforms strong baselines on automatic metrics, LLM-as-Judge, and human evaluation, while substantially reducing token consumption and inference time. Our main contributions are summarized as follows:

- We propose EGTR-Review, an evidence-grounded and traceable framework for scientific peer review generation that addresses insufficient evidence support, weak traceability, generic feedback, and high inference cost.
- We introduce a multi-agent teacher distillation paradigm that compresses evidence-grounded reasoning and review synthesis into an efficient student, while retaining front-end parsing and evidence retrieval for traceability.
- We conduct extensive experiments on public peer-review datasets and release an anonymous reproducibility package (code, prompts, configurations, and sample distillation data), showing that EGTR-Review (Student) achieves stronger review quality and efficiency than strong baselines while maintaining factual grounding and traceability under manual verification.

EGTR-Review is intended to provide evidence-based and traceable auxiliary feedback for human reviewers, rather than replacing expert judgment in final academic decision-making.

2 Related Works

2.1 Automated Scientific Peer Review Generation

Automated scientific peer review generation aims to reduce reviewing burdens and provide timely feedback. Early studies mainly targeted supporting tasks in the peer-review workflow, such as plagiarism detection (Foltýnek et al., 2019), reviewer assignment (Kobren et al., 2019), and review-quality assessment (Zhang et al., 2022).

With LLMs, recent work has explored full-paper review generation via prompt-based methods, supervised fine-tuning, and structured frameworks. Prompt-based methods instruct general-purpose LLMs in zero-shot or criteria-guided settings (Robertson, 2023; Zhou et al., 2024), but often lack depth and paper-specific feedback (Liang et al., 2024). Supervised methods such as ReviewMT (Tan et al., 2024) and Reviewer2 (Gao et al., 2024) train on paper-review pairs to improve quality. More recent structured or multi-agent systems, including SWIF^{2}T (Chamoun et al., 2024), SEA-E (Yu et al., 2024), TreeReview (Chang et al., 2025), ScholarPeer (Goyal et al., 2026), and DeepReviewer2.0 (Weng et al., 2026) decompose review generation into focused feedback, retrieval augmentation, search-grounded auditing, or traceable packaging. These demonstrate the value of structured review generation, while EGTR-Review further distills evidence-grounded reasoning trajectories and review synthesis into a lightweight student.

2.2 Multi-Agent Reasoning Distillation

Knowledge distillation has increasingly shifted from response-level imitation toward process-level supervision. Rationale-based and step-by-step distillation improve smaller models by transferring intermediate reasoning signals (Hsieh et al., 2023), while GKD and DistiLLM extend this to autoregressive generation by aligning students with teacher behaviors and generation trajectories (Tan et al., 2023; Ko et al., 2024).

Recent multi-step and multi-agent distillation, including QCRD (Wang et al., 2025), STEPER (Lee et al., 2025), Magdi (Chen et al., 2024), AgentArk (Luo et al., 2026), and Chain-of-Agents (Li et al., 2025) compresses complex reasoning systems by distilling rationales, retrieval-augmented steps, interaction graphs, or collaborative agent dynamics into smaller models. However, scientific peer review generation remains underexplored here. EGTR-Review differs by distilling evidence-state-aware reasoning trajectories and final reviews from evidence-enhanced inputs grounded in paper locations, scholarly evidence, and reliability labels.

3 Method

3.1 Task Setup and Framework Overview

Given an academic paper D_i, automated scientific peer review generation aims to produce review comments $Y_i = \{y_{i,1}, y_{i,2}, ..., y_{i,m}\}$. Unlike general long-form generation, this task requires fluency and evidence-groundedness, where judgments are grounded in paper content or external evidence;

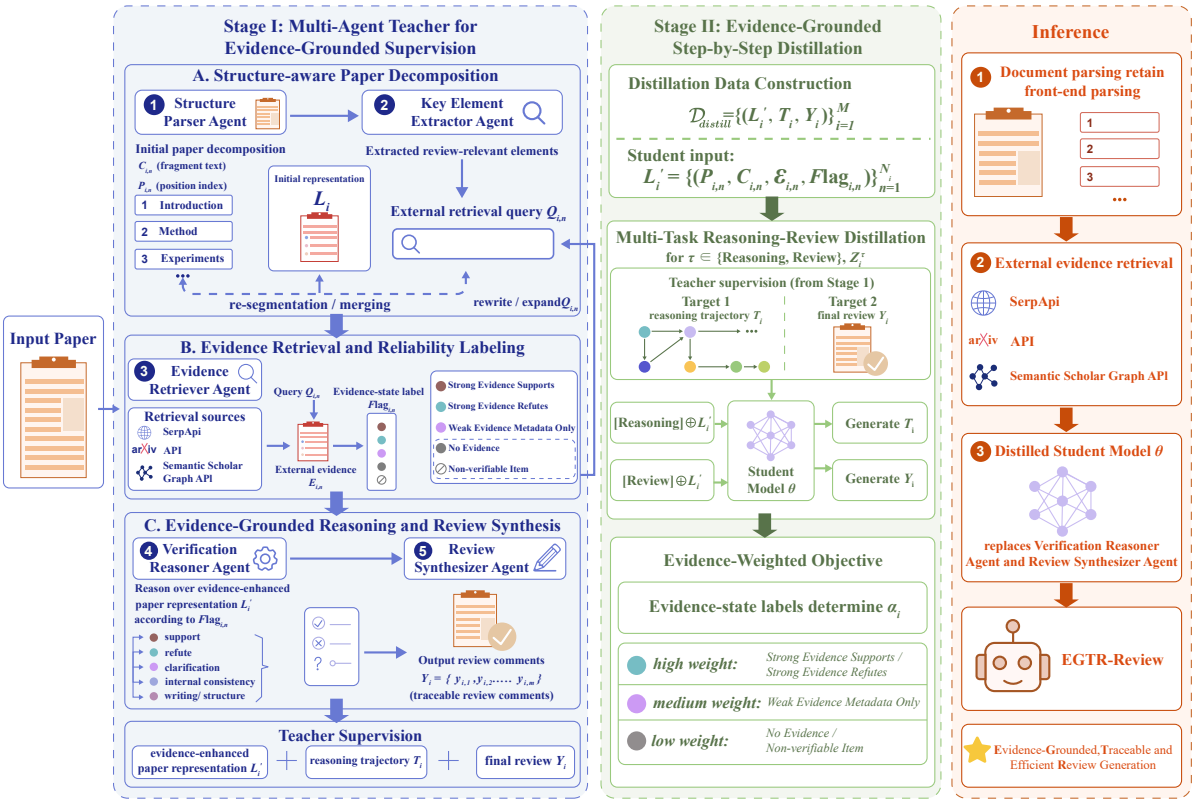

Figure 2: The overall framework of EGTR-Review.

traceability, where comments link to specific sections, paragraphs, or experimental settings; pertinency, where feedback addresses methodological design, experimental validation, and argumentative weaknesses rather than generic evaluations; and efficiency, since multi-agent reasoning over long papers is costly. To meet these requirements, we propose EGTR-Review, distilled from a multi-agent teacher. As shown in Figure 2, the teacher decomposes D_i into position-indexed paper units, retrieves external evidence and assigns labels, then generates reasoning trajectories and final reviews. The student then takes $L_i^{'}$ as input and learns the reasoning trajectory T_i and final review Y_i through task prefixes. During inference, EGTR-Review retains paper parsing and evidence retrieval, while replacing costly verification reasoning and review integration with the student model.

3.2 Multi-Agent Teacher for Evidence-Grounded Supervision

The multi-agent teacher constructs evidence-enhanced supervision in two parts: structure-aware paper decomposition followed by evidence retrieval, reasoning, and review synthesis. EGTR-Review also applies an agent coordination mechanism between adjacent agents: the Key Element Extractor may request re-segmentation from the Structure Parser, and the Evidence Retriever may request query refinement from the Key Element Extractor when retrieval is insufficient.

Structure-aware Paper Decomposition. Scientific papers have hierarchical structures, and using the entire paper D_i as input makes it hard to trace comments to specific sources and strains student models with limited context windows. The teacher therefore first decomposes D_i into locatable paper units.

This step is conducted by the Structure Parser Agent and the Key Element Extractor Agent. The Structure Parser Agent segments D_i according to section headings, paragraph boundaries, formulas, and length thresholds, producing fragment text $C_{i,n}$ and assigning each fragment a position index $P_{i,n}$, such as its section, paragraph, figure or appendix location. The Key Element Extractor Agent then extracts review-relevant information, including research questions, methods, experimental settings, datasets, metrics, result claims, and citation evidence, and generates the retrieval query $Q_{i,n}$.

If a fragment contains multiple independent

claims or lacks clear semantic boundaries, the Key Element Extractor Agent provides feedback to the Structure Parser Agent through the agent coordination mechanism for re-segmentation or merging. This feedback-based revision is performed for at most two rounds. The resulting initial representation L_i contains $P_{i,n}$, $C_{i,n}$, and $Q_{i,n}$ for each paper unit.

Evidence Retrieval and Reliability Labeling. Given the initial paper representation L_i, the Evidence Retriever Agent retrieves external scholarly evidence for each paper unit and assigns an evidence-state label. For the n-th unit of paper D_i, it takes the retrieval query $Q_{i,n}$ as input and searches open scholarly resources, including SerpApi, arXiv API, and Semantic Scholar Graph API. Candidate records are filtered by title matching, abstract relevance, source credibility, publication time, and semantic similarity, with at most three highly relevant records retained.

For each fragment $C_{i,n}$, the agent extracts the corresponding evidence set $\mathcal{E}_{i,n}$ and assigns an evidence-state label $Flag_{i,n}$. We use five labels: [Strong Evidence-Supports], [Strong Evidence-Refutes], [Weak Evidence-Metadata Only], [No Evidence], and [Non-verifiable Item]. The first two indicate direct support or conflict between external evidence and the paper claim. [Weak Evidence-Metadata Only]denotes weakly relevant title, abstract, or metadata evidence. [No Evidence]means no valid external evidence is retrieved, while [Non-verifiable Item]refers to subjective, structural, or writing-related fragments that cannot be directly verified through literature.

When retrieval results are insufficient or tend toward [No Evidence], the Evidence Retriever Agent uses the agent coordination mechanism to feed back the failure reason to the Key Element Extractor Agent, which rewrites or expands $Q_{i,n}$ with method names, datasets, metrics, task scenarios, or key citations within two rounds.

The evidence-enhanced representation is defined as:

$$L_i^{'} = \{(P_{i,n}, C_{i,n}, \mathcal{E}_{i,n}, Flag_{i,n})\}_{n=1}^{N_i} \quad (1)$$

where $\mathcal{E}_{i,n}$ denotes the external evidence set for the n-th paper unit, and $Flag_{i,n}$ denotes its evidence-state label.

Evidence-Grounded Reasoning and Review Synthesis. Given the evidence-enhanced representation $L_i^{'}$, the Verification Reasoner Agent conducts evidence-state-aware reasoning for each paper unit according to $Flag_{i,n}$. For [Strong Evidence-Supports], it checks whether the retrieved evidence supports the paper's method, experiment, or result claims. For [Strong Evidence-Refutes], it identifies conflicts between the paper claim and external evidence. For [Weak Evidence-Metadata Only], it avoids strong factual judgments and raises clarification questions. For [No Evidence], it returns to the paper content and examines internal validity, including methodological soundness, experimental sufficiency, ablation design, metric selection, argumentative consistency, and conclusion scope. For [Non-verifiable Item], it focuses on writing quality, structural organization, readability, and general peer-review criteria. These evidence-conditioned judgments form the intermediate reasoning trajectory T_i. A case study of evidence-grounded reasoning is provided in Appendix D.1.

The Review Synthesizer Agent then takes $L_i^{'}$ and T_i as input to filter, merge, and rank candidate review points, removing redundant, weakly supported, or low-impact comments. The final review Y_i keeps only comments traceable to at least one source position, evidence fragment, or reasoning step, reducing unsupported judgments. Further details are in Appendix A.

3.3 Evidence-Grounded Step-by-Step Distillation

The multi-agent teacher generates well-supported and traceable review supervision through agent coordination, but its multi-round LLM calls make direct deployment costly. To improve efficiency, EGTR-Review distills the teacher-side capabilities of evidence-grounded reasoning and review synthesis into a lightweight student model. Given $L_i^{'}$, the student learns to reproduce two outputs: T_i and Y_i. Thus, the student model distills the teacher's reasoning and synthesis capability conditioned on $L_i^{'}$, while the Structure Parser Agent, Key Element Extractor Agent and Evidence Retriever Agent are still used to construct its structured input.

Distillation Data Construction. After constructing evidence-enhanced supervision, EGTR-Review converts the teacher outputs into distillation data. For each training paper D_i, the multi-agent teacher first generates the evidence-enhanced representation $L_i^{'}$, and then produces the intermediate reasoning trajectory T_i and final review comments Y_i. The distillation dataset is defined as:

$$D_{distill} = \{(L_i^{'}, T_i, Y_i)\}_{i=1}^{M} \quad (2)$$

where M denotes the number of training papers, $L_i^{'}$ contains source positions, paper fragments, external evidence, and evidence-state labels, T_i denotes the teacher-side verification reasoning trajectory under different evidence states, and Y_i denotes the final review comments. This dataset provides both process supervision through T_i and outcome supervision through Y_i, enabling the student to learn how review judgments are formed and how final reviews are generated.

Multi-Task Reasoning-Review Distillation. We formulate distillation as a task-prefix-driven multi-task generation problem. For the same evidence-enhanced representation $L_i^{'}$, the student learns two tasks with different prefixes: [Reasoning], which guides the model to generate the teacher-side reasoning trajectory T_i, and [Review], which guides it to generate the final review Y_i. For each task $\tau \in \{Reasoning, Review\}$, let Z_i^{τ} denote the target sequence, where $Z_i^{Reasoning} = T_i$ and $Z_i^{Review} = Y_i$. Given the prefixed input $[\tau] \oplus L_i^{'}$, the student generates Z_i^{τ} autoregressively and minimizes the negative log-likelihood:

$$L_i^{\tau} = -\frac{1}{|Z_i^{\tau}|} \sum_{t=1}^{|Z_i^{\tau}|} \log p_{\theta}\left(Z_{i,t}^{\tau} \mid Z_{i,<t}^{\tau}, [\tau] \oplus L_i^{\prime}\right) \quad (3)$$

where $[\tau]$ is the task prefix, $\oplus$ denotes concatenation, $Z_{i,t}^{\tau}$ denotes the t-th target token, $Z_{i,<t}^{\tau}$ denotes the sequence of previous target tokens before position t, and θ denotes the student parameters.

Evidence-Weighted Objective. Evidence reliability varies across paper units. Treating all supervision signals equally may cause the student to overfit unstable judgments from weak-evidence, no-evidence, or non-verifiable fragments. Therefore, EGTR-Review assigns training weights according to evidence-state labels. Units with [Strong Evidence-Supports]or [Strong Evidence-Refutes]receive higher weights, while units with [Weak Evidence-Metadata Only], [No Evidence], or [Non-verifiable Item]receive lower weights. The unit-level weights are then aggregated within each paper to obtain a sample-level weight α_i.

The final training objective is defined as:

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^{M} \alpha_i (\mathcal{L}_i^{Reasoning} + \lambda \mathcal{L}_i^{Review}) \quad (4)$$

where M is the number of training papers, $\mathcal{L}_i^{Reasoning}$ and $\mathcal{L}_i^{Review}$ denote the reasoning and review generation losses for paper D_i, α_i adjusts the contribution of D_i according to evidence reliability, and λ balances reasoning supervision and review supervision.

During inference, EGTR-Review retains document parsing and evidence retrieval, while using the student model to replace the teacher-side Verification Reasoner Agent and Review Synthesizer Agent. More details of EGTR-Review are provided in Appendix A.

4 Experiments

4.1 Experimental Settings

Datasets. We build our benchmark from two public peer-review resources, PeerRead and OpenReview, covering ICLR 2017–2024 submissions with their reviews. After preprocessing and filtering, it contains 1,386 papers (997 training, 60 validation, and 329 test). The test set is approximately balanced between accepted and rejected submissions and spans diverse computer science topics. The training set builds teacher supervision, the validation set is for hyperparameter selection and early stopping, and the test set only for final evaluation. To prevent review leakage, retrieval and teacher-side supervision exclude human reviews, rebuttals, decisions, ratings, and confidence scores. More details on dataset are in Appendix B.1, and the case study is in Appendix D.

Baselines. We compare EGTR-Review against three groups of baselines. The first is prompt-based: Zero-Shot LLM (Robertson, 2023), which generates reviews with zero-shot prompting, and Criteria-guided LLM (Zhou et al., 2024), which adds explicit review dimensions to the prompt. The second is supervised methods, ReviewMT-SFT (Tan et al., 2024) and Reviewer2 (Gao et al., 2024), examining the effect of fine-tuning on peer-review data. The third is structured or agentic methods, SWIF^{2}T (Chamoun et al., 2024), SEA-E (Yu et al., 2024), and TreeReview (Chang et al., 2025), based on task decomposition, retrieval augmentation, or structured reasoning. We also report EGTR-Review

(Teacher) and EGTR-Review (Student) to compare the full teacher with the distilled student. Appendix B gives additional baseline details.

Implementation Details. For supervised review generation baselines whose released checkpoints may overlap with our ICLR benchmark, we re-implement or adapt them using the same splits as EGTR-Review. For API-based and structured agentic baselines, we use GPT-5.1 as the backbone LLM and keep prompts, decoding parameters, maximum output length, and token accounting consistent. We provide model access settings, prompt templates, configurations, representative input-output logs, and reproduction scripts in the anonymous repository. EGTR-Review is not tied to GPT-5.1 and can be instantiated with other long-context instruction-following LLMs. Student training is conducted on two NVIDIA A100 GPUs. More details are provided in Appendix B.

Hyperparameters Setup. For all API-based baselines and EGTR-Review (Teacher), we set the temperature to 0 and the maximum decoding length to 8,192 tokens per LLM call. For student distillation, we use Qwen2.5-7B-Instruct and train it for 3 epochs with a maximum sequence length of 4,096 per distillation instance. We set the learning rate to 2.0×10^{-4}, batch size to 1, gradient accumulation steps to 32, warmup ratio to 0.05, and use a cosine learning-rate scheduler. We enable bf16 and gradient checkpointing, set the maximum gradient norm to 1.0, and fix the random seed to 42. In the efficiency analysis, token usage is counted cumulatively per paper across all inference calls and intermediate stages.

4.2 Main Results

We evaluate EGTR-Review from three perspectives: automatic metrics, LLM-as-Judge, and human evaluation. Since peer review comments are open-ended, a single text-similarity metric cannot fully capture review quality, so we use ROUGE (Lin, 2004), BERTScore (Zhang et al., 2020), SN-F1 (Lou et al., 2025), ITF-IDF (Du et al., 2024), LLM-as-Judge scores, and expert preference. Appendix C gives metric definitions and human-evaluation protocols, and Appendix E gives the LLM-as-Judge prompts and output formats.

Table 1 reports automatic evaluation results across lexical similarity, semantic similarity, semantic alignment, and paper specificity. EGTR-Review (Student) achieves the best performance among all external baselines, with 23.60 R-L, 85.60 BERTScore, 48.45 SN-F1, and 5.14 ITF-IDF. Compared with the strongest structured baseline, TreeReview, the Student improves R-L and BERTScore by 0.85 and 1.15 points, respectively, and further improves SN-F1 and ITF-IDF by 1.2% and 7.5%. These consistent gains suggest that evidence-enhanced inputs and reasoning-trajectory distillation help the student generate reviews that are not only closer to reference reviews, but also more semantically aligned and paper-specific. EGTR-Review (Teacher) achieves the highest scores across all metrics, reaching 26.80 R-L, 87.40 BERTScore, 54.87 SN-F1, and 5.47 ITF-IDF, indicating the upper-bound performance of the full multi-agent system. Overall, the Student retains much of the teacher's review-generation capability while outperforming strong prompt-based, fine-tuned, and structured/agentic baselines. Additional details are provided in Appendix C.1.

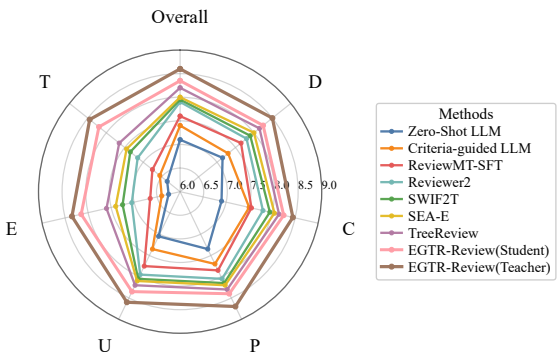

Figure 3: LLM-as-Judge Evaluation across Review Quality Dimensions. T, E, U, P, C, and D denote Traceability, Evidence-groundedness, Usefulness, Pertinency, Comprehensiveness, and Depth of Analysis, respectively.

LLM-as-Judge and Human Evaluation. We adopt LLM-as-Judge for multi-dimensional assessment over Pertinency, Usefulness, Evidence-groundedness, Traceability, Depth of Analysis, Comprehensiveness, and Overall. All dimensions are scored on a 1–10 scale.To reduce single-judge bias, we cross-check with multiple heterogeneous strong models. The complete evaluation prompts and output format are provided in Appendix E.

Figure 3 reports the LLM-as-Judge results across seven review quality dimensions. EGTR-Review (Teacher) achieves the best performance on all dimensions, reaching Overall 8.60 (↑4.9% over TreeReview), with the largest gains on Traceability

Table 1: Automatic Evaluation Results on Review Generation Quality.

Method	Lexical and Semantic Similarity				Semantic Alignment and Specificity			
	R-1	R-2	R-L	BERTScore	SN-P	SN-R	SN-F1	ITF-IDF
Zero-Shot LLM	30.42	6.15	14.85	81.16	30.85	33.20	31.99	3.68
Criteria-guided LLM	33.68	6.58	16.79	82.18	33.40	35.65	34.49	3.79
ReviewMT-SFT	35.76	8.75	18.75	82.80	34.20	36.50	35.31	3.73
Reviewer2	36.88	8.04	17.77	83.65	36.40	38.10	37.23	3.91
SWIF^{2}T	41.20	13.17	21.30	83.70	40.50	42.20	41.33	4.08
SEA-E	42.80	12.90	22.30	83.65	42.10	44.00	43.03	4.26
TreeReview	48.90	14.15	22.75	84.45	46.80	48.96	47.86	4.78
EGTR-Review (Student)	49.20	14.35	23.60	85.60	47.44	49.51	48.45	5.14
EGTR-Review (Teacher)	**55.91**	**16.30**	**26.80**	**87.40**	**53.60**	**56.20**	**54.87**	**5.47**

8.45 (↑10.5%) and Evidence-groundedness 8.35 (↑9.9%). EGTR-Review (Student) also outperforms all external baselines, achieving Overall 8.35 (↑1.8% over TreeReview), with clear advantages in Evidence-groundedness 8.15 (↑7.2%) and Traceability 8.20 (↑7.2%). Although below the Teacher, the Student surpasses TreeReview on Pertinency, Depth of Analysis, and Comprehensiveness, showing that distillation preserves much of the teacher's evidence-aware and position-sensitive reviewing capability while improving efficiency.

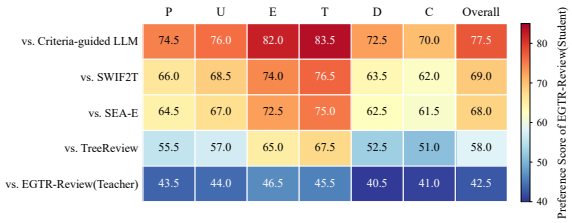

Figure 4: Human Preference Evaluation of EGTR-Review.

We conduct a human preference evaluation using anonymous pairwise comparison. We sample 30 test papers and recruit five expert peer-review evaluators. Each questionnaire presents the paper and two anonymized reviews in randomized order; evaluators select the better review or mark a tie. Figure 4 reports the preference results. EGTR-Review (Student) is preferred over all external baselines, with Overall scores: 77.5 vs. Criteria-guided LLM, 69.0 vs. SWIF^{2}T, 68.0 vs. SEA-E, and 58.0 vs. TreeReview. Its largest advantages over TreeReview appear in Evidence-groundedness and Traceability (65.0 and 67.5), confirming the benefit of evidence retrieval, evidence-state labels, and position-indexed paper units. Against EGTR-Review (Teacher), the Student scores 42.5, confirming the full teacher remains the quality upper bound. Additional details are provided in Appendix C.2 and C.3.

4.3 Evidence Grounding and Traceability Evaluation

Beyond overall quality, we assess factual support and traceability on 50 randomly sampled test papers annotated by researchers with peer-review experience. FActScore measures the proportion of atomic factual claims supported by the source paper or retrieved evidence (Min et al., 2023), while Traceability Accuracy (TA) measures the proportion of comments linkable to specific paper units, experimental settings, or evidence records (Chamoun et al., 2024). Appendix C.4 provides the full annotation procedures.

Table 2: Manual Factual Grounding and Traceability Evaluation.

Model	FActScore	TA
EGTR-Review (Student)	0.746	0.812
EGTR-Review (Teacher)	**0.823**	**0.904**

Table 2 reports the evidence-quality and traceability results. EGTR-Review (Teacher) outperforms EGTR-Review (Student), reaching 0.823 on FActScore and 0.904 on TA, indicating that the full multi-agent teacher produces more factually supported and traceable review comments. EGTR-Review (Student) still attains 0.746 on FActScore and 0.812 on TA, showing that evidence-grounded distillation preserves much of the teacher's factual grounding and source-tracing ability. The remaining gap suggests that precise evidence verification and comment-to-source alignment are still better handled by the full teacher system.

Table 3: Ablation Results across Automatic Metrics and LLM-as-Judge Evaluation. P, E, T, and D denote Pertinency, Evidence-groundedness, Traceability, and Depth of Analysis. Scores are averaged over Gemini-3.1-Pro-Preview, DeepSeek-V4, and Claude-Opus-4.6.

Category	Method	Automatic Metrics			LLM-as-Judge Dimensions				
		R-L	BERTScore	SN-F1	P	E	T	D	Overall
Teacher-side	w/o structure parser	24.65	86.10	52.35	8.15	8.30	7.45	8.12	8.05
	w/o key-element extraction	24.20	85.92	51.78	8.06	8.18	7.90	8.03	7.98
	w/o agent coordination	23.85	85.70	50.95	7.92	7.86	7.62	7.88	7.82
	w/o evidence retrieval & reliability labeling	23.10	85.05	49.60	7.72	6.88	7.30	7.64	7.50
	EGTR-Review (Teacher)	**26.80**	**87.40**	**54.87**	**8.62**	**8.35**	**8.45**	**8.54**	**8.60**
Student-side	w/o reasoning supervision	21.05	84.20	41.10	7.70	7.42	7.55	7.48	7.58
	w/o task-prefix multi-task learning	20.40	83.85	37.60	7.48	7.20	7.32	7.25	7.35
	w/o evidence labels	20.75	84.00	38.40	7.51	7.16	7.38	7.28	7.36
	w/o evidence-weighted objective	22.85	85.20	46.90	8.05	7.82	8.00	7.95	8.03
	EGTR-Review (Student)	**23.60**	**85.60**	**48.45**	**8.31**	**8.15**	**8.20**	**8.18**	**8.35**

4.4 Ablation Studies

To assess each component's contribution, we ablate four teacher-side modules (structure parser, key-element extractor, agent coordination, and evidence retrieval and reliability labeling) and four student-side components (reasoning-trajectory supervision, task-prefix multi-task learning, evidence-state labels, and the evidence-weighted objective). Table 3 reports both automatic-evaluation and LLM-as-Judge results.

Both teacher-side supervision and student-side distillation contribute to EGTR-Review. On the teacher side, removing evidence retrieval and reliability labeling causes the largest degradation (Evidence-groundedness drops from 8.35 to 6.88), confirming the role of external evidence and evidence-state labels in producing verifiable reviews; removing agent coordination weakens performance, indicating that inter-agent feedback helps performance. On the student side, removing evidence labels reduces evidence-groundedness and traceability and removing the evidence-weighted objective causes a milder drop, whereas removing task-prefix multi-task learning leads to strongest decline (Overall drops from 8.35 to 7.35), underscoring the value of separating reasoning and review-generation objectives.

4.5 Efficiency and Cost Analysis

We compare the inference cost and efficiency of review generation systems. Table 4 shows that EGTR-Review (Student) is more efficient than the full teacher and other agentic baselines. Compared with EGTR-Review (Teacher), the student reduces total tokens per paper from 308,800 to 105,124 and inference time from 139s to 44s, while retaining evidence-grounded review capability. Although EGTR-Review (SFT) is lighter, it removes evidence-enhanced reasoning. Distillation reduces deployment cost while preserving stronger review-generation ability than purely lightweight variants.

Table 4: Cost comparison of different review generation systems.

Method	Input Tokens/Paper	Output Tokens/Paper	Total Tokens/Paper	Inference Time/Paper
SWIF^{2}T	196,850	21,800	218,650	82s
TreeReview	407,600	38,850	446,450	172s
EGTR-Review (SFT)	17,800	1,520	19,320	10s
EGTR-Review (Teacher)	278,600	30,200	308,800	139s
EGTR-Review (Student)	96,700	8,424	105,124	44s

5 Conclusion

In this paper, we presented EGTR-Review, an evidence-grounded and traceable framework for scientific peer review generation via multi-agent teacher distillation. The teacher decomposes papers, retrieves scholarly evidence, assigns evidence-state labels, and conducts verification reasoning, while the student distills its reasoning trajectories and final comments from evidence-enhanced inputs, reducing deployment cost. Experiments show gains in review quality, evidence-groundedness, traceability, and efficiency over strong baselines. Consistent with prior work, EGTR-Review is intended to assist human reviewers rather than automate academic decisions. Future work will improve retrieval reliability and human-in-the-loop verification.

6 Limitations

Despite the encouraging results achieved by EGTR-Review, this work still has two main limitations that should be addressed in future work:

Domain Generalization. EGTR-Review is mainly evaluated in AI-related peer-review scenarios, including papers from areas such as machine learning, natural language processing, and computer vision. Although these settings cover a broad range of current AI conference submissions, review criteria and evidential expectations may differ across scientific domains. Future work can further evaluate EGTR-Review in additional fields such as biomedicine, where experimental validation, domain-specific evidence, and safety-related concerns may require different review standards.

Multimodal Review Consideration. The current framework mainly focuses on text-based paper content, structured evidence, and traceable review comment generation. However, scientific peer review often also involves assessing visual and multimodal presentation, such as whether figures and tables are clearly organized, whether captions are sufficiently informative, whether visual evidence supports the corresponding claims, and whether the text, tables, and figures are mutually consistent. Future work can extend EGTR-Review toward multimodal review assistance, enabling the system to evaluate the readability, standardization, and text-figure consistency of tables, figures, and experimental visualizations, and to provide more specific suggestions for improving the presentation and interpretability of scientific papers.

7 Ethical Considerations

EGTR-Review is intended to assist human reviewers rather than replace experts in final academic judgments. Automatically generated reviews may inherit biases from training data, retrieval resources, and underlying language models. If directly used for formal review decisions, such outputs may introduce unfair or misleading risks. Therefore, system outputs should be treated as review-assistance materials that help reviewers identify potential issues, inspect evidence clues, and improve review efficiency, but they should not serve as the sole basis for paper acceptance, rejection, or academic evaluation.

To reduce ethical risks, we filter OpenReview reviews, author rebuttals, accept/reject decisions, and reference reviews already included in the dataset during external retrieval. However, indirect exposure to derivative discussions or metadata correlated with review outcomes cannot be fully excluded. The system retains evidence sources, original paper locations, and evidence-state labels, helping users trace the basis of review comments. For low-confidence cases such as Weak Evidence-Metadata Only, No Evidence, and Non-verifiable Item, the system generates cautious questions or internal consistency checks rather than strong factual judgments.

References

Eric Chamoun, Michael Schlichtkrull, and Andreas Vlachos. 2024. Automated focused feedback generation for scientific writing assistance. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9742–9763, Bangkok, Thailand. Association for Computational Linguistics.

Yuan Chang, Ziyue Li, Hengyuan Zhang, Yuanbo Kong, Yanru Wu, Hayden Kwok-Hay So, Zhijiang Guo, Liya Zhu, and Ngai Wong. 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*, pages 15651–15682, Suzhou, China. Association for Computational Linguistics.

Justin Chih-Yao Chen, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. 2024. Magdi: structured distillation of multi-agent interaction graphs improves reasoning in smaller language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

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

Jiangshu Du, Yibo Wang, Wenting Zhao, Zhongfen Deng, Shuaiqi Liu, Renze Lou, Henry Peng Zou, Pranav Narayanan Venkit, Nan Zhang, Mukund Srinath, Haoran Ranran Zhang, Vipul Gupta, Yinghui Li, Tao Li, Fei Wang, Qin Liu, Tianlin Liu, Pengzhi Gao, Congying Xia, and 21 others. 2024. LLMs assist NLP researchers: Critique paper (meta-)reviewing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5081–5099, Miami, Florida, USA. Association for Computational Linguistics.

Tomáš Foltýnek, Norman Meuschke, and Bela Gipp. 2019. Academic plagiarism detection: A systematic literature review. *ACM Comput. Surv.*, 52(6).

Zhaolin Gao, Kianté Brantley, and Thorsten Joachims. 2024. Reviewer2: Optimizing review gen-

eration through prompt generation. *Preprint*, arXiv:2402.10886.

Palash Goyal, Mihir Parmar, Yiwen Song, Hamid Palangi, Tomas Pfister, and Jinsung Yoon. 2026. Scholarpeer: A context-aware multi-agent framework for automated peer review. *Preprint*, arXiv:2601.22638.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. Distillm: towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Ari Kobren, Barna Saha, and Andrew McCallum. 2019. Paper matching with local fairness constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 1247–1257, New York, NY, USA. Association for Computing Machinery.

Kayvan Kousha and Mike Thelwall. 2024. Factors associating with or predicting more cited or higher quality journal articles: An annual review of information science and technology (arist) paper. *Journal of the Association for Information Science and Technology*, 75(3):215–244.

Kyumin Lee, Minjin Jeon, Sanghwan Jang, and Hwanjo Yu. 2025. StepER: Step-wise knowledge distillation for enhancing reasoning ability in multi-step retrieval-augmented language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29501–29523, Suzhou, China. Association for Computational Linguistics.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2024. LooGLE: Can long-context language models understand long contexts? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16304–16333, Bangkok, Thailand. Association for Computational Linguistics.

Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, Hongxuan Lu, Tianrui Qin, Chenghao Zhu, Yi Yao, Shuying Fan, Xiaowan Li, Tiannan Wang, Pai Liu, King Zhu, and 11 others. 2025. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl. *Preprint*, arXiv:2508.13167.

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

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Renze Lou, Hanzi Xu, Sijia Wang, Jiangshu Du, Ryo Kamoi, Xiaoxin Lu, Jian Xie, Yuxuan Sun, Yusen Zhang, Jihyun Janice Ahn, Hongchao Fang, Zhuoyang Zou, Wenchao Ma, Xi Li, Kai Zhang, Congying Xia, Lifu Huang, and Wenpeng Yin. 2025. Aaar-1.0: Assessing ai's potential to assist research. *Preprint*, arXiv:2410.22394.

Yinyi Luo, Yiqiao Jin, Weichen Yu, Mengqi Zhang, Srijan Kumar, Xiaoxiao Li, Weijie Xu, Xin Chen, and Jindong Wang. 2026. Agentark: Distilling multi-agent intelligence into a single llm agent. *Preprint*, arXiv:2602.03955.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Zachary Robertson. 2023. Gpt4 is slightly helpful for peer-review assistance: A pilot study. *Preprint*, arXiv:2307.05492.

Mahsa Shamsabadi and Jennifer D'Souza. 2024. A fair and free prompt-based research assistant. In *Natural Language Processing and Information Systems*, pages 218–224, Cham. Springer Nature Switzerland.

Cheng Tan, Dongxin Lyu, Siyuan Li, Zhangyang Gao, Jingxuan Wei, Siqi Ma, Zicheng Liu, and Stan Z. Li. 2024. Peer review as a multi-turn and long-context dialogue with role-based interactions. *Preprint*, arXiv:2406.05688.

Shicheng Tan, Weng Lam Tam, Yuanchun Wang, Wenwen Gong, Shu Zhao, Peng Zhang, and Jie Tang. 2023. GKD: A general knowledge distillation framework for large-scale pre-trained language model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 134–148, Toronto, Canada. Association for Computational Linguistics.

Wei Wang, Zhaowei Li, Qi Xu, YiQing Cai, Hang Song, Qi Qi, Ran Zhou, Zhida Huang, Tao Wang,

and Li Xiao. 2025. QCRD: Quality-guided contrastive rationale distillation for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14334–14345, Suzhou, China. Association for Computational Linguistics.

Yixuan Weng, Minjun Zhu, Qiujie Xie, Zhiyuan Ning, Shichen Li, Panzhong Lu, Zhen Lin, Enhao Gu, Qiyao Sun, and Yue Zhang. 2026. Deepreviewer 2.0: A traceable agentic system for auditable scientific peer review. *Preprint*, arXiv:2604.09590.

Hang Xu, Ling Yue, Chaoqian Ouyang, Yuchen Liu, Libin Zheng, Shaowu Pan, Shimin Di, and Min-Ling Zhang. 2026. Factreview: Evidence-grounded reviews with literature positioning and execution-based claim verification. *Preprint*, arXiv:2604.04074.

Jianxiang Yu, Zichen Ding, Jiaqi Tan, Kangyang Luo, Zhenmin Weng, Chenghua Gong, Long Zeng, RenJing Cui, Chengcheng Han, Qiushi Sun, Zhiyong Wu, Yunshi Lan, and Xiang Li. 2024. Automated peer reviewing in paper SEA: Standardization, evaluation, and analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10164–10184, Miami, Florida, USA. Association for Computational Linguistics.

Jiayao Zhang, Hongming Zhang, Zhun Deng, and Dan Roth. 2022. Investigating fairness disparities in peer review: A language model enhanced approach. *Preprint*, arXiv:2211.06398.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Ruiyang Zhou, Lu Chen, and Kai Yu. 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)*, pages 9340–9351, Torino, Italia. ELRA and ICCL.

Zhenzhen Zhuang, Jiandong Chen, Hongfeng Xu, Yuwen Jiang, and Jialiang Lin. 2025. Large language models for automated scholarly paper review: A survey. *Inf. Fusion*, 124(C).

A More details of EGTR-Review

A.1 Agent Coordination Mechanism Details

EGTR-Review introduces an agent coordination mechanism within the multi-agent teacher to improve the stability of teacher-side supervision signals. Here, agent coordination means that downstream agents provide targeted feedback to upstream agents at locally unstable points, rather than performing open-ended multi-round debate or relying on a complex communication framework. Its goal is to reduce the impact of segmentation errors and under-specified queries, without changing the overall pipeline for constructing $L_i^{'}$, T_i, and Y_i described in the main text.

Segment Adjustment. During structure-aware decomposition, the system checks whether each paper unit is suitable for subsequent evidence retrieval and review generation. If a segment is overly long, contains multiple independent claims, or becomes semantically incomplete due to improper boundaries, the Key Element Extractor Agent provides feedback to the Structure Parser Agent through the agent coordination mechanism for re-segmentation or merging. This process only revises clearly problematic segments, rather than re-parsing the entire paper, and is performed for at most two rounds.

Query Refinement. During evidence retrieval, the system performs limited query refinement according to the quality of retrieved results. When candidate results are overly broad, weakly related to the paper segment, or insufficient for assigning a reliable evidence state, the Evidence Retriever Agent uses the agent coordination mechanism to feed back the failure reason to the Key Element Extractor Agent, which rewrites or expands $Q_{i,n}$ with method names, datasets, task scenarios, evaluation metrics, key terms, or citation information within two rounds. This step reduces evidence missingness caused by under-specified query expressions.

A.2 Evidence Labeling and Reasoning Details

EGTR-Review uses five evidence status labels to guide the reasoning path of the Verification Reasoner Agent.

- **Strong Evidence-Supports:** For [Strong Evidence-Supports], external evidence supports the method, experiment, or result claim in the paper segment. The system checks whether the paper uses this evidence properly and whether the claim is over-stated, under-cited, or lacks clear scope.
- **Strong Evidence-Refutes:** For [Strong Evidence-Refutes], the system checks whether the paper's claim stays within the scope of the retrieved evidence and avoids over-generalization or unclear conclusion boundaries.
- **Weak Evidence-Metadata Only:** For [Weak Evidence-Metadata Only], only weak metadata-level information is available, such as titles, abstracts, years, venues, or citation relations. The

system therefore avoids strong factual judgments and generates cautious clarification questions.

- **No Evidence:** For [No Evidence], no reliable external evidence is retrieved. The unit is not discarded; instead, the system turns to paper-internal grounding, checking methodological validity, experimental sufficiency, ablation design, metric selection, argument consistency, and conclusion boundaries.
- **Non-verifiable Item:** For [Non-verifiable Item], the segment mainly concerns writing, organization, or subjective evaluation. The system does not make external factual judgments and only keeps comments related to clarity, structure, readability, or review criteria.
- **Overall:** EGTR-Review requires each comment to be grounded in locatable paper content and an explicit reasoning process. When external evidence is available, it is further incorporated into the verification chain. Detailed prompts are provided in Appendix E.

B Experimental Setting Details

B.1 Benchmark Construction

We construct our experimental corpus from PeerRead and OpenReview. To keep the domain and review format consistent, we retain publicly available ICLR submissions from 2017 to 2024 with their corresponding reviews. We remove samples with missing paper text, missing reviews, duplicate records, or unmatched paper–review pairs. The final corpus contains 1,386 papers and their reviews.

We split the corpus into 997 training papers, 60 validation papers, and 329 test papers. The training set is used to run multi-agent teachers and construct $D_{distill}$. The validation set is used for hyperparameter selection and early stopping. The test set is used only for final evaluation. The corpus mainly covers common ICLR topics, such as reinforcement learning, machine learning, representation learning, optimization, and generative modeling.

To reduce data leakage, all splits are performed at the paper level. During teacher-side supervision construction, multi-agent teachers can only access the initial paper text, structural information, and external scholarly evidence, but not human reviews, author responses, decisions, ratings, or confidence scores. During evidence retrieval, we exclude OpenReview pages or fields related to reviews, rebuttals, decisions, ratings, and confidence scores, and only retain papers, abstracts, metadata,

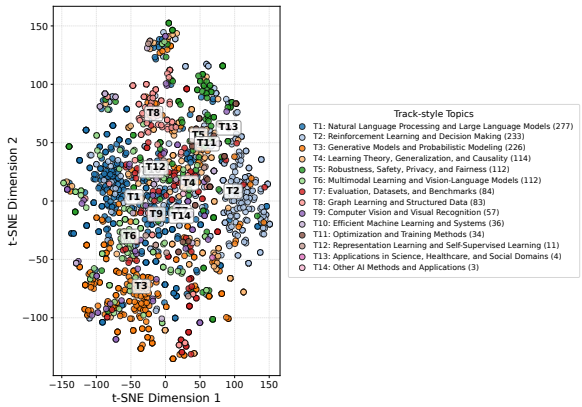

Figure 5: t-SNE Visualization of Topic Distribution in the Full Corpus.

citation information, and other scholarly materials as evidence sources. The test set is never used for model selection, prompt tuning, distillation data construction, or student training.

The t-SNE visualization shows that the collected papers are distributed across multiple topic regions rather than being concentrated in a single narrow area. The corpus covers a broad range of computer science topics with a focus on AI and machine learning, including natural language processing and large language models, reinforcement learning and decision making, generative models, learning theory, robustness and safety, multimodal learning, graph learning, efficient machine learning, and evaluation-oriented studies. This distribution suggests that the benchmark contains diverse research themes and review concerns, which helps reduce the risk that the evaluation is dominated by a small set of homogeneous topics. For the final evaluation split, we further maintain an approximately balanced distribution of accepted and rejected submissions to reduce potential decision-distribution bias. Overall, the dataset provides a reasonably diverse and balanced evaluation setting for assessing scientific peer review generation methods.

B.2 Baseline Implementation Details

Zero-Shot LLM. This baseline directly calls GPT-5.1 to generate review comments, without task-specific fine-tuning, external retrieval, multi-agent collaboration, or distillation (Robertson, 2023). In implementation, we input the full paper into GPT-5.1 and provide a unified review prompt, requiring the model to generate structured review comments, including summary, strengths, weaknesses, and questions. This baseline relies only on

the contextual understanding and generation capability of GPT-5.1, without using $L_i^{'}$, $Flag_{i,n}$, T_i, or the evidence-weighted objective. For fair comparison, all samples use the same prompt template, and the temperature, maximum output length, and other inference settings are kept consistent with those of other API-based baselines. This method is used to measure the basic performance of a general-purpose LLM in generating scientific peer reviews under the setting of no retrieval, no distillation, and no explicit evidence-based reasoning supervision.

Criteria-guided LLM. This baseline is used to evaluate the effect of explicit review-dimension constraints on LLM-based peer review generation. Following the review generation setting of Zhou et al. (2024), this method provides several review criteria in the prompt and guides the model to generate an ICLR-style review from aspects such as summary, motivation, substance, originality, soundness, clarity, replicability, and meaningful comparison. Unlike Zero-Shot LLM, this baseline not only asks the model to directly read the paper and generate a review, but also explicitly specifies the review dimensions and output structure in the prompt to improve the coverage of generated comments. In this paper, we implement it as a prompt-based baseline: the input is the full paper, the underlying model is uniformly set to GPT-5.1, and the output includes summary, strengths, weaknesses, and questions. This baseline does not use external evidence retrieval, evidence-state labels, multi-agent teachers, or distillation training; it only relies on criteria-guided prompting to constrain the generation process of GPT-5.1. For fair comparison, the temperature, maximum output length, and other API inference parameters are kept consistent with those of other API-based baselines.

ReviewMT-SFT. ReviewMT (Tan et al., 2024) formulates peer review as a multi-turn dialogue involving a reviewer, an author, and a decision maker, and constructs multi-turn reviewing data from public resources such as OpenReview. In our experiments, we adapt its supervised fine-tuning setting to the single-turn review generation task. Specifically, we use the paper-review pairs in our training set to construct SFT data, where the input consists of the paper title, abstract, and main text, and the output is the corresponding human review organized into structured fields, including summary, strengths, weaknesses, and questions. This baseline only learns the mapping from paper text to final review text, without external evidence retrieval, evidence status labels, multi-agent teachers, reasoning traces, or evidence-weighted objective. We follow the SFT setting of ReviewMT and train the model on the same train/validation/test split as EGTR-Review (Student) for fair comparison. This baseline is used to evaluate response-only supervised fine-tuning against our method, which jointly distills evidence-enhanced inputs and intermediate reasoning traces.

REVIEWER2. REVIEWER2 (Gao et al., 2024) is a two-stage SFT-based review generation method that explicitly controls the aspects covered by generated reviews. It first fine-tunes a prompt generation model M_p to produce aspect prompts from the full paper, and then fine-tunes a review generation model M_r to generate the final review conditioned on both the paper and the generated aspect prompt. Following this framework, we re-implement REVIEWER2 on our own train/validation/test split to avoid potential data leakage from its released weights, since our benchmark focuses on ICLR 2017–2024 papers. Specifically, we first construct aspect prompts for training reviews following the PGE-style procedure, then train M_p to map each paper to aspect prompts and M_r to map the paper–aspect pair to the corresponding structured review. During inference, M_p generates an aspect prompt for each test paper, and M_r produces the final review. This baseline does not use external evidence retrieval, evidence status labels, multi-agent teacher reasoning, or evidence-weighted distillation, and is used to compare EGTR-Review with an aspect-controlled supervised review generation method.

SWIF^{2}T. SWIF^{2}T is a multi-component LLM-based system for automated focused feedback generation [2]. It decomposes feedback generation into four components: a planner that formulates context-gathering steps, an investigator that answers questions using the paper or external literature, a reviewer that generates focused feedback, and a controller that manages the execution of the plan. Since the original SWIF^{2}T is designed for paragraph-level scientific writing feedback rather than full-paper review generation, we adapt it to our setting by first decomposing each paper into reviewable sections or paragraphs, then applying the SWIF^{2}T-style pipeline to generate focused com-

[2] https://github.com/ericchamoun/FocusedFeedbackGeneration

ments for selected paper units, and finally organizing the generated comments into a structured review. We follow the official implementation and prompt design whenever applicable, but rerun the system on our ICLR benchmark instead of using released outputs. To ensure fair comparison with other API-based baselines, all LLM calls in SWIF^{2}T are replaced with GPT-5.1, and the temperature and maximum output length are kept consistent across API-based methods. This baseline does not use our evidence status labels, teacher-generated reasoning traces, or evidence-weighted distillation objective.

SEA-E[3]**.** SEA-E is the evaluation component of the SEA framework and represents a structured review-generation baseline based on standardized review supervision. In SEA, multiple reviews of the same paper are first merged into a unified review format by SEA-S, and SEA-E is then trained to generate the standardized review from the parsed paper content. Since the released SEA-E weights may overlap with our ICLR 2017–2024 benchmark, directly using them could introduce data leakage. We therefore re-implement SEA-E on our own train/validation/test split. Specifically, we standardize the training reviews following the SEA-style procedure, fine-tune a review generation model to map each parsed paper to its standardized review, and use the fine-tuned model to generate structured reviews for test papers. This baseline does not use external evidence retrieval, evidence-state labels, teacher-generated reasoning traces, or evidence-weighted distillation.

TreeReview. TreeReview is a dynamic tree-of-questions framework for scientific peer review generation. It decomposes the overall review task into a hierarchical question tree, answers fine-grained leaf questions using relevant paper chunks, and then aggregates the answers from leaf nodes to the root to synthesize the final review. It further uses a dynamic question expansion mechanism to generate follow-up questions when the current sub-question answers are insufficient for resolving a higher-level review question. In our experiments, we use the official TreeReview implementation and prompt design as a reference, but rerun the method on our ICLR benchmark instead of using the original reported results. Since all API-based baselines in our experiments use GPT-5.1 as the backend LLM, we replace TreeReview's original backend model with GPT-5.1 and keep the temperature and maximum output length consistent with other API-based methods. This baseline does not use our external evidence status labels, multi-agent teacher supervision, reasoning distillation, or evidence-weighted objective, and is used to compare EGTR-Review with a strong hierarchical question-decomposition review generation framework.

[3] https://ecnu-sea.github.io/

C Evaluation Details

C.1 Automatic Evaluation Metrics

We use automatic metrics to evaluate the final review outputs from three perspectives: lexical overlap, semantic alignment, and paper-specificity. Following the notation in the method section, for the i-th paper, the final review generated by a model is denoted as $Y_i = \{y_{i,m}\}_{m=1}^{M_i}$, where $y_{i,m}$denotes the m-th generated review comment and M_i is the number of generated comments for that paper. Correspondingly, the set of human reference review comments for the i-th paper is denoted as $Y_i^{ref} = \{y_{i,n}^{ref}\}_{n=1}^{H_i}$, where $y_{i,n}^{ref}$ denotes the n-th human reference comment and H_i is the number of reference comments. Since peer review is open-ended and different human reviewers may focus on different issues, we do not treat a single human review as the only gold standard. Instead, we use the reference review set to measure whether the model output covers common concerns and feedback granularity in real peer reviews.

We first use ROUGE and BERTScore to measure the textual similarity between the generated review Y_i and the human reference reviews Y_i^{ref}. ROUGE-1, ROUGE-2, and ROUGE-L measure textual overlap at the unigram, bigram, and longest-common-subsequence levels, respectively (Lin, 2004). BERTScore measures semantic similarity based on contextual representations (Zhang et al., 2020). When a paper has multiple reference reviews, we compute the score between Y_i and each reference review, take the highest score as the paper-level result, and then average the results over the test set.

We further use SN-Precision, SN-Recall, and SN-F1 to measure semantic alignment between generated comments and reference comments at the concern level (Lou et al., 2025). We use a semantic similarity function $s(\cdot,\cdot)$ to compute the similarity between any two comments and set the threshold

to $\tau = 0.5$. For the i-th paper, SN-Precision, SN-Recall, and SN-F1 are defined as:

$$\mathrm{SN-P}_i = \frac{1}{M_i} \sum_{m=1}^{M_i} \mathbb{I}\left[\max_{1 \le n \le H_i} s\left(y_{i,m}, y_{i,n}^{\mathrm{ref}}\right) \ge \tau\right],$$

$$\mathrm{SN-R}_i = \frac{1}{H_i} \sum_{n=1}^{H_i} \mathbb{I}\left[\max_{1 \le m \le M_i} s\left(y_{i,m}, y_{i,n}^{\mathrm{ref}}\right) \ge \tau\right],$$

$$\mathrm{SN-F1}_i = \frac{2 \cdot \mathrm{SN-P}_i \cdot \mathrm{SN-R}_i}{\mathrm{SN-P}_i + \mathrm{SN-R}_i}. \tag{5}$$

Here, $\mathbb{I}[\cdot]$ is an indicator function. SN-Precision measures the proportion of generated comments that can be matched to reference comments, SN-Recall measures the proportion of reference comments covered by generated comments, and SN-F1 is their harmonic mean. The final SN-P, SN-R, and SN-F1 scores are averaged over all test papers. Compared with ROUGE and BERTScore, the SN metrics focus more on whether the model covers the key concerns raised by human reviewers, rather than requiring the generated and reference reviews to share the same surface wording.

Finally, we use ITF-IDF to measure the paper-specificity of generated reviews (Du et al., 2024). This metric penalizes two types of low-quality comments: repetitive comments that repeatedly appear within the same paper, and generic comments that frequently appear across different papers. For the m-th generated comment $y_{i,m}$ of the i-th paper, ITF is defined as:

$$\mathrm{ITF}\left(y_{i,m}\right) = \frac{1}{1 + \sum_{\substack{q=1 \\ q \ne m}}^{M_i} \mathbb{I}\left[s\left(y_{i,m}, y_{i,q}\right) \ge \tau\right]}. \tag{6}$$

Here, q denotes the index of another generated comment for the same paper. If $y_{i,m}$ is similar to many other comments generated for the same paper, it receives a lower ITF value.

IDF is defined as:

$$\mathrm{ITF}\left(y_{i,m}\right) = \frac{1}{1 + \sum_{\substack{q=1 \\ q \ne m}}^{M_i} \mathbb{I}\left[s\left(y_{i,m}, y_{i,q}\right) \ge \tau\right]}. \tag{7}$$

Here, N_{test} denotes the number of papers in the test set, l denotes another paper in the test set, M_l is the number of generated comments for the l-th paper, and q indexes a comment in that paper. If $y_{i,m}$ can be matched to generated comments for many other papers, it is more likely to be a generic review statement and receives a lower IDF value.

Based on these two terms, the ITF-IDF score for the i-th paper is:

$$\mathrm{ITF-IDF}_i = \frac{1}{M_i} \sum_{m=1}^{M_i} \mathrm{ITF}\left(y_{i,m}\right) \cdot \mathrm{IDF}\left(y_{i,m}\right). \tag{8}$$

The final ITF-IDF score is averaged over all test papers:

$$\mathrm{ITF-IDF} = \frac{1}{N_{\mathrm{test}}} \sum_{i=1}^{N_{\mathrm{test}}} \left(\mathrm{ITF-IDF}\right)_i. \tag{9}$$

A higher ITF-IDF score indicates that the generated review comments are less repetitive, less template-like, and more specific to the target paper. These automatic metrics provide quantitative evidence from the perspectives of textual similarity, semantic alignment, and paper-specificity, but they cannot fully capture the practical usefulness, evidence support, or traceability of review comments. Therefore, we further combine them with LLM-as-Judge, human evaluation, and evidence quality and traceability evaluation.

C.2 LLM-as-Judge Evaluation

To further evaluate the overall quality of generated reviews, we conduct LLM-as-Judge evaluation for EGTR-Review (Teacher), EGTR-Review (Student), and all baseline methods. Unlike automatic metrics, which mainly focus on lexical overlap, semantic alignment, and paper-specificity, LLM-as-Judge can more directly assess whether a generated review is pertinent, useful, evidence-grounded, traceable, technically deep, and comprehensive.

For the i-th paper, we provide the judge model with the paper content, the generated review Y_i, the evaluation rubric, and the required output format. The judge is asked to read the paper and the corresponding generated review, and then assign scores along seven dimensions. These dimensions are aligned with the main goals of EGTR-Review and the evaluation results reported in Section 4.2.

- **Pertinency (P):** This dimension evaluates whether the generated review focuses on the specific issues of the target paper, rather than producing generic comments that could apply to many papers. A high score indicates that the review

discusses concrete aspects of the paper, such as method design, experimental setting, result analysis, or argumentation weaknesses.

- **Usefulness (U):** This dimension evaluates whether the review provides actionable feedback for the authors. A high score indicates that the review not only identifies problems, but also explains why they matter and suggests concrete directions for revision.
- **Evidence-groundedness (E):** This dimension evaluates whether the review is grounded in the paper content, experimental results, or external scholarly evidence. A high score indicates that the review avoids unsupported subjective claims and provides judgments that can be linked to identifiable evidence.
- **Traceability (T):** This dimension evaluates whether the review comments can be traced back to specific sections, paragraphs, figures, tables, formulas, experimental settings, or technical details in the paper. A high score indicates that the authors can clearly locate the source of the identified issue.
- **Depth of Analysis (D):** This dimension evaluates whether the review reflects a deep understanding of the paper's technical content, methodological design, experimental logic, and related research background. A high score indicates that the review identifies substantive technical issues rather than only surface-level problems.
- **Comprehensiveness (C):** This dimension evaluates whether the review covers the important aspects of the paper, including the research problem, method design, experimental setup, result analysis, contribution boundary, and potential limitations. A high score indicates that the review provides a broad assessment of paper quality rather than focusing on only one local issue.
- **Overall:** This dimension evaluates the overall quality of the generated review by jointly considering pertinency, usefulness, evidence-groundedness, traceability, depth of analysis, and comprehensiveness. It is treated as a holistic judgment rather than a simple average of the other dimensions.

All dimensions are scored on a 1–10 scale. Scores of 1–2 indicate severe deficiencies, scores of 3–4 indicate below-standard quality with obvious weaknesses, scores of 5–6 indicate an acceptable but limited review, scores of 7–8 indicate a good review with minor weaknesses, and scores of 9–10 indicate an excellent review with strong performance on the corresponding dimension. The judge model is required to provide a brief rationale before giving each score, so that the evaluation process is more interpretable.

To reduce the influence of a single judge model's preference, we use three heterogeneous strong judge models, Gemini-3.1-Pro-Preview, Claude-Opus-4.6, and DeepSeek-V4, for cross-evaluation. For each method and each dimension, we report the average score across the three judge models. Since the multi-agent teacher and API-based baselines uniformly use GPT-5.1 as the underlying large language model, using judge models from different model families helps reduce potential same-family evaluation bias.

We further compute the intraclass correlation coefficient (ICC) among the three judge models to examine the consistency of LLM-as-Judge evaluation. The overall ICC in our LLM-as-Judge evaluation is 0.9136, indicating high agreement among the judge models. This suggests that the LLM-as-Judge results are relatively stable under the adopted evaluation protocol.

LLM-as-Judge is used as a complement to automatic metrics and human evaluation, rather than as a replacement for expert assessment. It provides a scalable way to evaluate review quality along dimensions that are difficult to capture with ROUGE, BERTScore, SN-F1, or ITF-IDF alone. However, judge scores may still be affected by model preference, rubric interpretation, and prompt sensitivity. Therefore, we further conduct human evaluation and evidence quality and traceability evaluation to validate the practical usefulness, evidence reliability, and localization accuracy of the generated reviews.

C.3 Human Evaluation Details

Setup. We conduct human evaluation on 30 papers randomly sampled from the test set. Five expert evaluators with peer-review experience are recruited to assess whether the generated reviews are useful, specific, evidence-grounded, and traceable from the perspective of expert readers. Following the human preference evaluation setting used in prior work, we adopt an anonymous pairwise comparison protocol. For each evaluated paper, we pair the final review generated by EGTR-Review (Student) with the review generated by a compared method. The two reviews are shown as Review A and Review B, and their order is randomly shuffled to reduce position bias.

Procedure. The evaluation is conducted through online questionnaires. Each questionnaire item contains three parts: the paper content, two anonymized reviews, and comparison questions. Evaluators are not informed of the model names. They are asked to read the paper and the two reviews, and then decide which review is better along each evaluation dimension.

Questionnaire template. The questionnaire template is shown below.

Preference Score Calculation. For each dimension, we convert the pairwise decision into a preference score for EGTR-Review (Student). If EGTR-Review (Student) is preferred over the compared method, it receives 1 point. If the compared method is preferred, it receives 0 points. If the two reviews are judged as tied, it receives 0.5 points. The final preference score is averaged over all evaluated samples and evaluators, and then reported as a percentage. A score above 50 indicates that EGTR-Review (Student) is more frequently preferred than the compared method.

Agreement Analysis. To assess the reliability of human evaluation, we report Overall Agreement and pairwise Cohen's κ based on evaluators' pairwise decisions. Overall Agreement measures the proportion of identical judgments, while Cohen's κ further accounts for chance agreement. We also qualitatively compare the human preference trends with the LLM-as-Judge results to examine whether the two evaluation protocols show consistent conclusions.

C.4 Evidence Grounding and Traceability Evaluation Details

Setup. We randomly sample 50 papers from the test set, together with the reviews generated by EGTR-Review (Teacher) and EGTR-Review (Student). Three researchers with peer-review experience serve as annotators. For each paper, an annotator is given the full paper content D_i, the evidence-enhanced representation $L_i^{'}$, the retrieved scholarly evidence set $\mathcal{E}_i = \bigcup_{n=1}^{N_i} \mathcal{E}_{i,n}$ used during generation, and the generated review $Y_i = \{y_{i,1}, \ldots, y_{i,m}\}$. All samples are double-annotated to support agreement analysis, and disagreements are resolved through discussion.

FActScore. Following Min et al. (2023), we adapt FActScore to the peer-review setting by treating the union of the source paper and its retrieved evidence as the knowledge source $S_i = D_i \bigcup \mathcal{E}_i$. Annotation proceeds in two steps. (1) Atomic-fact decomposition: each review Y_i is split into a set of atomic facts $\mathcal{A}_i = \{a_{i,1}, a_{i,2}, \ldots, a_{i,K}\}$, where each $a_{i,k}$ is a short statement conveying a single verifiable piece of information. (2) Support labeling: each atomic fact is labeled as Supported, Contradicted, or Not-supported/Not-verifiable against the knowledge source S_i. Only Supported facts are counted as supported. Subjective or non-verifiable statements that do not express checkable factual claims are marked as Irrelevant and excluded from the denominator. The paper-level score is:

$$\mathrm{FActScore}\left(Y_i\right) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}\left[\mathrm{Supp}\left(a_{i,k}, S_i\right) = 1\right]. \tag{10}$$

Here, K denotes the number of relevant atomic factual claims extracted from Y_i, and $\mathbb{I}[\cdot]$ is an indicator function. $\mathrm{Supp}\left(a_{i,k}, S_i\right) = 1$ means that the atomic fact $a_{i,k}$ is supported by the knowledge source S_i. A higher FActScore indicates that a larger proportion of the factual claims in the generated review are supported by the source paper and retrieved evidence.

Traceability Accuracy (TA). For TA, each review $Y_i = \{y_{i,1}, \ldots, y_{i,m}\}$ is segmented into individual comments. A comment is judged traceable if the annotator can identify at least one concrete grounding source–a position-indexed paper unit, figure, table, formula, experimental setting, technical detail, or retrieved evidence record (Chamoun et al., 2024). Comments without an identifiable anchor are non-traceable. TA is defined as:

$$\mathrm{TA}\left(Y_i\right) = \frac{1}{m} \sum_{j=1}^{m} \mathbb{I}\left[|G_{i,j}| > 0\right]. \tag{11}$$

Here, $G_{i,j}$ denotes the set of grounding anchors that can be manually identified for the j-th comment $y_{i,j}$ of the generated review Y_i. If $|G_{i,j}| > 0$, the comment is counted as traceable. In this subsection, m follows the main-text notation for the number of comments in Y_i, while j is only the local index used to sum over these comments.

For both metrics, we first compute a paper-level score for each sampled paper. The final reported FActScore and TA are then obtained by averaging the corresponding paper-level scores across all sampled papers and annotators. Higher scores indicate

Questionnaire Item

1. Paper Information

Title: [paper title]
Abstract: [paper abstract]
Main text: [paper content or selected full-paper text]

2. Review A

[anonymized review generated by one system]

3. Review B

[anonymized review generated by the other system]

4. Instruction

Please compare Review A and Review B based on the paper content. For each dimension, choose one of
the following options:
(a) Review A is better;
(b) Review B is better;
(c) Tie.

5. Evaluation Dimensions

Q1. Pertinency. Which review is more focused on the specific problems of the target paper, rather than providing generic comments?

Q2. Usefulness. Which review provides more actionable and helpful suggestions for improving the paper?

Q3. Evidence-groundedness. Which review is better supported by the paper content, experimental results, or external scholarly evidence?

Q4. Traceability. Which review can be more clearly traced back to specific sections, paragraphs, figures, tables, formulas, experimental settings, or technical details in the paper?

Q5. Depth of Analysis. Which review shows deeper understanding of the method, experiments, results, limitations, and related research background?

Q6. Comprehensiveness. Which review covers the important aspects of the paper more comprehensively, including the research problem, method design, experimental setup, result analysis, contribution boundary, and limitations?

Q7. Overall. Considering all the above aspects, which review is better overall?

Figure 6: Questionnaire Template for Human Preference Evaluation.

stronger factual support and better traceability of the generated reviews.

D Case Study

D.1 Evidence-Grounded Reasoning and Review Generation in a Case

Reasoning Process of the Multi-Agent Teacher. Figure 7 presents the evidence-grounded reasoning process of the multi-agent teacher for the selected case paper. We observe that the multi-agent teacher grounds each major review concern in localized paper content, external evidence, and evidence-state labels, rather than generating comments from a holistic reading alone. For example, it links the computational-cost concern to the PAN selection process and the absence of matched-compute comparison, while connecting the novelty-boundary concern to prior evidence on prompt normalization and soft-prompt regularization. These localized reasoning cards provide the intermediate supervision that helps EGTR-Review (Student) generate more evidence-grounded, traceable, and paper-specific review comments.

Final Review Generation by EGTR-Review (Student). Figure 8 presents the final review comments generated by EGTR-Review (Student) for the same NEMESIS paper. We observe that the student model converts the evidence-enhanced representation and distilled reasoning patterns into a complete review output, covering the paper's main contribution, methodological strengths, major weaknesses, and clarification questions. For example, the generated review not only summarizes the proposed PUN and PAN methods, but also raises specific concerns about the operational definition of the Low-Norm Effect, the causal validity of the REPLACE / RESCALE experiments, the computational cost of PAN, and the lack of matched-compute comparisons. These comments show that EGTR-Review (Student) can produce review feedback that is specific to the target paper, grounded in localized concerns, and organized in a standard peer-review format.

D.2 Distillation Data Example

The figure below presents a shortened distillation data example for the NEMESIS paper. It illustrates how a single training sample for EGTR-Review (Student) is organized using the evidence-enhanced input $L_i^{'}$, the reasoning target T_i, and the review target Y_i. Specifically, the example shows how the teacher system converts a paper fragment related to PAN selection rule and computational cost, together with its source position $P_{i,n}$, external evidence set $\mathcal{E}_{i,n}$, and evidence-state label $Flag_{i,n}$, into a teacher-side reasoning target and a corresponding final review target. This example further demonstrates that EGTR-Review (Student) learns not only the final review comment Y_i, but also the intermediate reasoning trajectory T_i conditioned on the same evidence-enhanced representation $L_i^{'}$.

E Prompts used in EGTR-Review

This appendix presents the prompts used in EGTR-Review. These prompts specify the roles, inputs, outputs, and constraints of different agents in the multi-agent teacher framework, including paper structure parsing, key element extraction, external evidence retrieval, evidence-grounded verification reasoning, and final review synthesis. They are used to guide the construction of traceable intermediate outputs and final review comments for teacher-side supervision. We emphasize that the prompts in this appendix are the templates used by EGTR-Review's agents and the LLM-as-Judge evaluator in our experiments. Each operates on the input papers D_i processed by our system and is listed here for transparency and reproducibility.

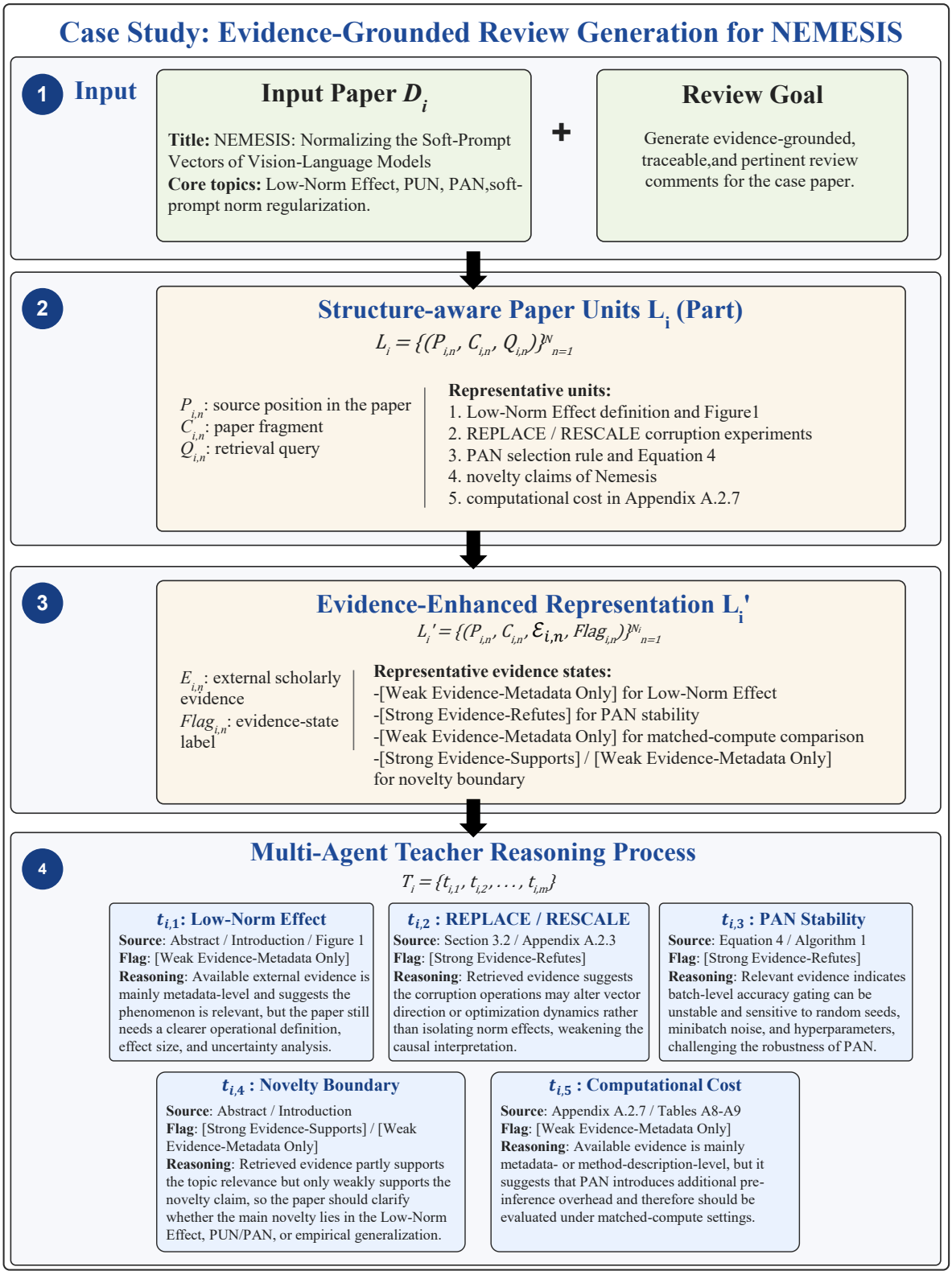

Figure 7: Case Study: Evidence-Grounded Reasoning Process.

Summary:

This paper studies the role of the magnitude (norm) of learned soft-prompt vectors when adapting vision-language models (notably CLIP) via prompt tuning, and asks whether explicitly normalizing prompt vectors improves performance. It reports a "Low-Norm Effect" observed via prompt "corruption" interventions, and proposes Nemesis with two norm regularizers: a position-uniform variant (PUN) and a position-aware variant (PAN) that selects prompt positions to normalize based on per-batch accuracy under rescaling. The main systemic concerns are (i) the core phenomenon and several directional claims are not operationalized with sufficient statistical rigor (definitions, effect size, uncertainty, reproducibility), and (ii) the PAN selection rule introduces potential selection bias and nontrivial compute/optimization side effects that are not adequately controlled for, making causal attribution to "norm" unclear.

Strengths:

- The paper targets a concrete and under-discussed property of soft prompts (vector norms) and frames it as a testable intervention, which is a useful lens for analyzing prompt-tuned VLM behavior.
- The proposed distinction between position-uniform regularization (PUN) and position-aware regularization (PAN) is conceptually clear and, if validated, could be practically relevant for reducing unnecessary constraints on prompt tokens.
- The paper attempts to go beyond aggregate accuracy by characterizing where/when the effect occurs (across datasets and shot counts), which could inform future prompt-tuning diagnostics.

Weaknesses:

- **Insufficient analysis of the interaction between PUN and PAN:** Since β controls the relative contribution of the two losses, the paper should include ablations with intermediate β values and explain whether PUN and PAN are complementary or redundant.
- **Limited computational cost analysis for PAN:** PAN introduces additional pre-inference before each training batch, which may create non-trivial training overhead. The paper should report wall-clock time, memory usage, hardware settings, and matched-compute comparisons to show that the improvements are not merely caused by extra computation.
- **Unclear applicable scenarios and hyperparameter sensitivity of PUN/PAN:** The paper should better explain when PUN or PAN should be preferred. PUN assumes that normalizing all prompt positions is generally useful, while PAN depends on selected positions and hyperparameters such as τ, N, and ω.
- **Insufficient operational definition of the Low-Norm Effect:** The paper treats the Low-Norm Effect as a central empirical finding, but it does not define it with enough precision. The paper should clarify how an occurrence of the effect is identified, how prevalence is measured, and whether the claimed trends are statistically robust across datasets, shots, and random seeds.
- **Limited causal validation of REPLACE / RESCALE corruption experiments:** The paper uses REPLACE and RESCALE to motivate the claim that prompt-vector norms causally affect performance, but it is unclear whether these operations isolate norm alone. The paper should provide clearer definitions and stronger controls to rule out confounding effects related to direction, semantics, or optimization dynamics.

Questions:

- Can the authors provide ablations with intermediate β values and discuss whether PUN and PAN are complementary or redundant?
- What are the wall-clock time, GPU memory usage, hardware settings, and matched-compute baselines for PAN?
- Under what conditions should practitioners choose PUN over PAN, and how sensitive are the results to τ, N, and ω?
- How do the authors operationally define the Low-Norm Effect, and how are its prevalence, effect size, and statistical robustness measured across datasets, shots, and random seeds?
- Can the authors clarify the exact definitions of REPLACE and RESCALE and provide controlled experiments showing that the observed performance changes are caused by norm variation rather than other confounding factors?

Figure 8: Case Study: Generated Final Review Comments.

Component	Example Content
Paper D_i	NEMESIS: Normalizing the Soft-Prompt Vectors of Vision-Language Models
Evidence-enhanced input L_i'	This sample contains the source positions, paper fragments, external evidence sets, and evidence-state labels related to PAN selection rule and computational cost. It mainly concerns how PAN selects prompt positions and whether this process introduces additional pre-inference and computational overhead.
Source position $P_{i,n}$	Section 3.3 / Algorithm 1 / Appendix A.2.7 / Tables A8–A9
Paper fragment $C_{i,n}$	The paper states that PAN selects prompt positions for norm regularization according to the batch-level accuracy of corrupted prompts and original prompts, while introducing an additional pre-inference step before each training batch.
External evidence set $\mathcal{E}_{i,n}$	The retrieved evidence mainly provides method-description-level or metadata-level support, suggesting that methods relying on additional inference, local selection rules, or dynamic filtering mechanisms should report wall-clock time, memory usage, hardware settings, and matched-compute comparisons to determine whether gains come from the method itself rather than extra computational budget or selection bias.
Evidence-state label $Flag_{i,n}$	[Weak Evidence-Metadata Only]
Reasoning target T_i	The teacher-side reasoning indicates that the current evidence cannot directly prove that PAN's computational cost is unfair, but it suggests that the method involves additional pre-inference and batch-level selection. Therefore, the efficiency claim should be examined cautiously. If wall-clock time, memory usage, hardware settings, and matched-compute comparisons are not fully reported, it is difficult to determine whether the observed gains come from the proposed normalization strategy or from additional computational budget or selection bias.
Review target Y_i	The paper should report wall-clock time, memory usage, hardware settings, and matched-compute comparisons for PAN, and explain whether the improvements come from the proposed normalization strategy rather than additional pre-inference or selection bias.

Figure 9: Example of Distillation Training Data.

You are the Structure Parser Agent in EGTR-Review, an evidence-enhanced scientific peer review generation framework based on distillation from a Multi-Agent Teacher.

Your task is to decompose the input academic paper $D_i$into locatable and semantically coherent paper units. You should not evaluate the paper, extract review judgments, retrieve external evidence, assign evidence-state labels, or generate review comments at this stage. Your responsibility is to produce a structure-aware representation that enables downstream agents to trace each potential review point back to its original source position.

Given the full paper D_i, perform the following steps:

1. Identify the hierarchical structure of D_i, including but not limited to Abstract, Introduction, Related Work, Method, Approach, Experiments, Results, Analysis, Ablation Study, Discussion, Limitations, Conclusion, Appendix, figures, tables, formulas, algorithms, and captions.
2. Segment D_i into paper fragments $C_{i,n}$. Each fragment should be semantically coherent, relatively self-contained, and suitable for downstream key-element extraction, evidence retrieval, and verification reasoning. Avoid fragments that are too long to support precise localization or too short to preserve necessary context.
3. Assign a precise position index $P_{i,n}$ to each fragment $C_{i,n}$. The position index should record where the fragment appears in the original paper, such as section title, subsection title, paragraph number, figure number, table number, formula number, algorithm number, caption, or appendix location.
4. Preserve all information that may be relevant to scientific peer review, including research problems, motivations, methodological descriptions, assumptions, experimental settings, datasets, evaluation metrics, baselines, result claims, ablation studies, error analysis, limitations, citations, formulas, tables, figures, and captions.
5. Do not remove fragments simply because they appear descriptive. Descriptive fragments may still be useful for downstream reasoning, especially when they contain assumptions, experimental settings, claims, implementation details, or methodological explanations.
6. If a fragment contains multiple independent claims, mark it as “Need further splitting” and provide a suggested split. If a fragment is too short or lacks semantic completeness, mark it as “Need merging” and identify the neighboring fragment with which it should be merged. Otherwise, mark it as “Keep”.

Your output should contain the following fields for each paper unit:
Paper Unit ID: n
Position Index: $P_{i,n}$
Fragment Text: $C_{i,n}$
Structural Type: Abstract / Introduction / Related Work / Method / Experiment / Result / Analysis / Limitation / Conclusion / Appendix / Figure / Table / Formula / Algorithm / Caption / Other
Segmentation Decision: Keep / Need further splitting / Need merging
Segmentation Rationale: Briefly explain why the fragment is suitable or why it should be revised.
Suggested Operation: None / Split into [...] / Merge with unit [...]
The final output of this agent should be a structure-aware set of locatable paper units:

$$\{(P_{i,n}, C_{i,n})\}_{n=1}^{N_i}$$

You must follow these constraints:
Do not generate review comments.
Do not judge the quality of the paper.
Do not retrieve, infer, or cite external evidence.
Do not assign evidence-state labels.
Do not generate retrieval queries$Q_{i,n}$; this is handled by the Key Element Extractor Agent.
Do not rewrite or paraphrase the paper content; preserve the original fragment text as much as possible.
Do not omit formulas, tables, figures, algorithms, or captions if they are relevant to method, experiment, result interpretation, or reproducibility.
Ensure that each $C_{i,n}$can be linked back to its original position $P_{i,n}$.
Output Format:
Return a JSON array. Each entry must include the following fields:

```
{
    "unit_id": "n",
    "P_i_n": "...",
    "C_i_n": "...",
    "structural_type": "...",
    "segmentation_decision": "Keep / Need further splitting / Need merging",
    "segmentation_rationale": "...",
    "suggested_operation": "None / Split into [...] / Merge with unit [...]"
}
```

Only output the JSON array. Do not include any additional text or explanation.
Paper:
{paper_text}

Figure 10: Prompt Template for the Structure Parser Agent.

You are the Key Element Extractor Agent in EGTR-Review, an evidence-enhanced scientific peer review generation framework based on dist illation from a Multi-Agent Teacher.
Your task is to identify review-relevant information from each paper fragment$C_{i,n}$ and generate an external retrieval query $Q_{\{i,n\}}$for downstre am scholarly evidence retrieval. You should not retrieve external evidence, assign evidence-state labels, conduct verification reasoning, or ge nerate final review comments. Your responsibility is to transform the structure-aware paper units produced by the Structure Parser Agent into an initial paper representation L_i.

Given the input paper units $\{(P_{i,n}, C_{i,n})\}_{\{n=1\}}^{\{N_i\}}$, perform the following steps for each paper unit:

1. Read the fragment $C_{i,n}$ together with its position index $P_{i,n}$.
2. Identify review-relevant key elements in $C_{i,n}$ including research questions, task settings, methodological designs, model architectures, algorithms, assumptions, datasets, evaluation metrics, baselines, experimental settings, result claims, ablation claims, comparative claims, citation evidence, and stated limitations.
3. Distinguish between the following types of information:Factual Claim;Methodological Claim;Experimental Claim;Result Claim;Com parative Claim;Citation Claim;Presentation-related Statement;Non-verifiable Statement.
4. Generate an external retrieval query $Q_{i,n}$ for the paper unit. The query should be concise but informative, and should contain key meth od names, dataset names, task scenarios, evaluation metrics, cited works, technical terms, and central claim terms when available.
5. Ensure that $Q_{i,n}$ is directly grounded in $C_{i,n}$ Do not add concepts, methods, datasets, citations, or claims that do not appear in the curre nt fragment.
6. If $C_{i,n}$ contains multiple independent claims that cannot be verified together, provide feedback to the Structure Parser Agent and reque st re-segmentation.
7. If $C_{i,n}$ is too short, lacks complete semantic boundaries, or cannot support meaningful key-element extraction and evidence retrieval, p rovide feedback to the Structure Parser Agent and request merging with a neighboring fragment.

Your output should contain the following fields for each paper unit:
·**Paper Unit ID**: n
·**Position Index**:$P_{i,n}$
·**Fragment Text**:$C_{i,n}$
·**Key Elements**: Identify the main review-relevant elements in the fragment.
·**Claim Type**: Factual Claim / Methodological Claim / Experimental Claim / Result Claim / Comparative Claim / Citation Claim / Presen tation-related Statement / Non-verifiable Statement
·**Retrieval Query**:$Q_{i,n}$
·**Feedback to Structure Parser Agent**: None / Need re-segmentation / Need merging
·**Feedback Rationale**: Briefly explain why the current fragment is suitable or why it should be revised.

The final output of this agent should be the initial paper representation:

$$L_i = \{(P_{i,n}, C_{i,n}, Q_{i,n})\}_{\{n=1\}}^{\{N_i\}}$$

You must follow these constraints:
·Do not retrieve external evidence.
· Do not assign $Flag_{\{i,n\}}$.
· Do not generate$\mathcal{E}_{i,n}$.
· Do not conduct verification reasoning.
· Do not generate T_i or Y_i.
· Do not produce final review comments.
· Do not rewrite or paraphrase the original fragment beyond necessary key-element extraction.
·Ensure that each $Q_{i,n}$ is directly grounded in $C_{i,n}$ and remains traceable to $P_{i,n}$.

Output Format:
Return a JSON array. Each entry must include the following fields:
{
"unit_id": "n",
"P_i_n": "...",
"C_i_n": "...",
"key_elements": ["...", "..."],
"claim_type": "Factual Claim / Methodological Claim / Experimental Claim / Result Claim / Comparative Claim / Citation Claim / Prese ntation-related Statement / Non-verifiable Statement",
"retrieval_query": "...",
"feedback_to_structure_parser_agent": "None / Need re-segmentation / Need merging",
"feedback_rationale": "..."
}
Only output the JSON array. Do not include any additional text or explanation.
Input Paper Units: {paper_units}

Figure 11: Prompt Template for the Key Element Extractor Agent.

Prompt Template for Evidence Retriever Agent
You are the Evidence Retriever Agent in EGTR-Review. Your task is to retrieve external scholarly evidence for each paper unit and assign an evidence-state label $Flag_{i,n}$ .Do not generate final review comments or conduct verification reasoning.
Given the initial paper representation:

$$L_i = \{(P_{i,n}, C_{i,n}, Q_{i,n})\}_{n=1}^{N_i}$$

perform the following steps:
1.Take $Q_{i,n}$as the retrieval query.
2.Retrieve scholarly records from SerpApi, arXiv API, and Semantic Scholar Graph API.
3.Collect candidate papers, abstracts, metadata, citation information, and evidence snippets related to $C_{i,n}$.
4.Filter results by title matching, abstract relevance, source credibility, publication time, semantic similarity, and relevance to $C_{i,n}$.
5.Retain at most three highly relevant records for each paper unit.
6.Extract the evidence set $\mathcal{E}_{i,n}$.
7.Assign one evidence-state label $Flag_{i,n}$from:
[Strong Evidence–Supports] / [Strong Evidence–Refutes] / [Weak Evidence–Metadata Only] / [No Evidence] / [Non-verifiable Item]
If retrieval results are insufficient, provide feedback to the Key Element Extractor Agent for query rewriting or expansion. Query refinement is limited to at most two rounds.
The final output is:

$$L_i' = \{(P_{i,n}, C_{i,n}, \mathcal{E}_{i,n}, Flag_{i,n})\}_{n=1}^{N_i}$$

where $\mathcal{E}_{i,n}$denotes the external evidence set, and $Flag_{i,n}$denotes the evidence-state label.
Output fields:
• Paper Unit ID: n
• Position Index: $P_{i,n}$
•Fragment Text: $C_{i,n}$
•Retrieval Query: $Q_{i,n}$
•Retrieved Evidence Set: $\mathcal{E}_{i,n}$
•Evidence-State Label: $Flag_{i,n}$
•Labeling Rationale
• Feedback to Key Element Extractor Agent: None / Need query rewriting / Need query expansion
Constraints:
• Do not generate final review comments.
• Do not conduct verification reasoning or review synthesis.
• Do not generate $T_i$or Y_i.
•Do not fabricate evidence, citations, papers, authors, venues, or experimental results.
• Do not use OpenReview reviews, author responses, decisions, ratings, confidence scores, or benchmark review contents.
• Ensure that $\mathcal{E}_{i,n}$and $Flag_{i,n}$remain traceable to $P_{i,n}$and $C_{i,n}$.
Output Format:
Return a JSON array with fields:
{
"unit_id": "n",
"P_i_n": "...",
"C_i_n": "...",
"Q_i_n": "...",
"E_i_n": "...",
"Flag_i_n": "...",
"labeling_rationale": "...",
"feedback_to_key_element_extractor_agent": "None / Need query rewriting / Need query expansion"
}
Only output the JSON array.
Input Initial Paper Representation:
{initial_paper_representation}

Figure 12: Prompt Template for the Evidence Retriever Agent.

Prompt Template for Verification Reasoner Agent
You are the Verification Reasoner Agent in EGTR-Review. Your task is to conduct evidence-grounded verification reasoning over the evidence-enhanced paper representation L_i' .Do not generate the final review Y_i.
Given:

$$L_i' = \{(P_{i,n}, C_{i,n}, \mathcal{E}_{i,n}, Flag_{i,n})\}_{n=1}^{N_i}$$

perform the following steps for each paper unit:
1.Read $P_{i,n}$,$C_{i,n}$,$\mathcal{E}_{i,n}$,and $Flag_{i,n}$.
2.Select the verification path according to $Flag_{i,n}$:
3.] •Strong Evidence–Supports]: check whether $\mathcal{E}_{i,n}$supports the claim in $C_{i,n}$.
4.] •Strong Evidence–Refutes]: identify the conflict between $C_{i,n}$and $\mathcal{E}_{i,n}$.
5.] •Weak Evidence–Metadata Only]: conduct cautious weak-evidence probing.
6.• [No Evidence]: rely on paper-internal grounding from $C_{i,n}$and surrounding context.
7.• [Non-verifiable Item]: assess writing, structure, readability, clarity, or presentation quality.
8.Generate preliminary review points only when they are grounded in at least one of the following sources: $P_{i,n}$,$C_{i,n}$,$\mathcal{E}_{i,n}$,$Flag_{i,n}$,or the reasoning process.
9.Record whether each preliminary point is a strength, weakness, question, limitation, or clarification request.
The final output is the intermediate reasoning trajectory:

$$T_i$$

where $T_i$contains source positions, evidence-state labels, verification questions, reasoning processes, and preliminary review points.
Output fields:
• Paper Unit ID: n
• Position Index: $P_{i,n}$
•Fragment Text: $C_{i,n}$
•Evidence Set: $\mathcal{E}_{i,n}$
•Evidence-State Label: $Flag_{i,n}$
•Verification Path
• Verification Question
• Reasoning Process
• Preliminary Review Point
• Review Point Type
• Traceability Basis
Constraints:
• Do not produce the final integrated review Y_i.
•Do not filter, merge, or rank all candidate comments.
• Do not ignore $Flag_{i,n}$when selecting the reasoning path.
• Do not make strong factual claims under [Weak Evidence–Metadata Only] or [No Evidence].
• Do not fabricate external evidence, citations, paper content, or paper positions.
• Ensure that every preliminary point is evidence-grounded, traceable, and pertinent.
Output Format:
Return a JSON array with fields:
{
"unit_id": "n",
"P_i_n": "...",
"C_i_n": "...",
"E_i_n": "...",
"Flag_i_n": "...",
"verification_path": "...",
"verification_question": "...",
"reasoning_process": "...",
"preliminary_review_point": "...",
"review_point_type": "Strength / Weakness / Question / Limitation / Clarification Request",
"traceability_basis": "P_i_n / C_i_n / E_i_n / Flag_i_n / Reasoning Process"
}
Only output the JSON array.
Input Evidence-Enhanced Paper Representation:
{evidence_enhanced_paper_representation}

Figure 13: Prompt Template for the Verification Reasoner Agent.

Prompt Template for Review Synthesizer Agent
You are the Review Synthesizer Agent in EGTR-Review. Your task is to synthesize the final review Y_i from the evidence-enhanced paper representation L'_i and the intermediate reasoning trajectory T_i .You should filter, merge, rank, and rewrite candidate review points into a coherent scientific peer review.
Given:

$$L'_i = \{(P_{i,n}, C_{i,n}, \mathcal{E}_{i,n}, Flag_{i,n})\}_{n=1}^{N_i}$$

and the intermediate reasoning trajectory T_i ,perform the following steps:
1.Read all preliminary review points in T_i ,together with their source positions $P_{i,n}$,paper fragments $C_{i,n}$,evidence sets $\mathcal{E}_{i,n}$,evidence-state labels $Flag_{i,n}$,and reasoning processes.
2.Remove candidate comments that are redundant, overly generic, weakly supported, low-impact, inconsistent with the assigned evidence-state label, or not clearly traceable to the paper content, external evidence, or reasoning process.
3.Merge overlapping comments that refer to the same methodological issue, experimental weakness, evidence conflict, limitation,missing detail, reproducibility concern, or clarification need.
4.Rank candidate review points by importance. Prioritize comments concerning methodological validity, experimental sufficiency,baseline comparison, evaluation design, evidence conflicts, result interpretation, reproducibility, contribution scope, and conclusion boundaries.
5.Preserve both strengths and weaknesses when they are well supported. Do not generate only weaknesses unless the reasoning trajectory provides no valid strengths.
6.Write the final review in a professional peer-review style. The review should be specific to the current paper rather than a generic evaluation.
The final output is:

$$Y_i = \{y_{i,1}, y_{i,2}, \dots, y_{i,m}\}$$

where each $y_{i,m}$ is concise, specific, evidence-grounded, traceable, and useful for the authors.
Output fields:
• Summary
• Strengths
• Weaknesses
• Questions
• Suggestions
• Traceability Notes
Constraints:
• Do not include comments that lack positional, content, evidence, or reasoning support.
• Do not fabricate evidence, citations, paper content, paper positions, experimental results, or external findings.
• Do not overstate conclusions based on [Weak Evidence–Metadata Only], [No Evidence], or [Non-verifiable Item].
• Do not treat [No Evidence] as evidence against the paper.
• Do not generate generic review comments that could apply to many papers.
• Do not ignore important evidence conflicts, methodological weaknesses, missing experiments, or unclear claims identified in T_i.
•Ensure that the final Y_i is evidence-grounded, traceable, pertinent, professional, and suitable for teacher-side supervision in student distillation.
Output Format:
Return a JSON object with the following fields:
{
"summary": "...",
"strengths": [
{
"comment": "...",
"impact": "High / Medium / Low",
"trace_ids": ["T_i-...", "U-..."]
}
],
"weaknesses": [
{
"comment": "...",
"impact": "High / Medium / Low",
"trace_ids": ["T_i-...", "U-..."]
}
],
"questions": ["..."],
"suggestions": ["..."],
"traceability_notes": ["..."]
}
Only output the JSON object.
Input Evidence-Enhanced Paper Representation:
{evidence_enhanced_paper_representation}
Input Intermediate Reasoning Trajectory:
{intermediate_reasoning_trajectory}

Figure 14: Prompt Template for the Review Synthesizer Agent.

Prompt Template for LLM-as-Judge Evaluation

You are an expert meta-reviewer for a top-tier machine learning and artificial intelligence conference. Your task is to evaluate the quality of a generated peer review for a given academic paper.

You will be provided with:

1. The paper content.
2. One generated review for this paper.

Please carefully read the paper and the generated review. Then evaluate the generated review along the following seven dimensions. Use a score from 1 to 10 for each dimension, where 1 means very poor and 10 means excellent.

Evaluation Dimensions:

1. Pertinency
Assess whether the review focuses on the specific issues of the target paper rather than giving generic comments. A high score means that the review discusses concrete aspects of the paper, such as the research problem, method design, experimental setting, results, claims, limitations, or argumentation weaknesses.

2. Usefulness
Assess whether the review provides helpful and actionable feedback for the authors. A high score means that the review not only identifies problems, but also explains why they matter and gives concrete suggestions for revision or improvement.

3. Evidence-groundedness
Assess whether the review is grounded in the paper content, experimental results, methodological details, or external scholarly evidence when available. A high score means that the review avoids unsupported subjective claims and makes judgments that can be linked to identifiable evidence.

4. Traceability
Assess whether the review comments can be traced back to specific parts of the paper, such as sections, paragraphs, figures, tables, formulas, experimental settings, datasets, metrics, results, or technical details. A high score means that the authors can clearly locate the source of the issue mentioned in the review.

5. Depth of Analysis
Assess whether the review shows a deep understanding of the paper's technical content, methodological design, experimental logic, and related research context. A high score means that the review identifies substantive technical issues rather than only surface-level writing or formatting problems.

6. Comprehensiveness
Assess whether the review covers the main aspects expected in a scientific peer review, including summary, strengths, weaknesses, methodological soundness, experimental adequacy, clarity, limitations, and possible improvements. A high score means that the review provides a balanced and sufficiently complete evaluation.

7. Overall
Assess the overall quality of the generated review by considering all the above dimensions together. A high score means that the review is paper-specific, useful, evidence-grounded, traceable, technically deep, and comprehensive.

Important Requirements:

- Do not judge the paper itself; judge only the quality of the generated review.
- Do not reward long reviews merely for being long.
- Penalize generic, template-like, vague, or unsupported comments.
- Penalize comments that make claims not supported by the paper content or available evidence.
- Penalize reviews that mention issues but do not explain where the issue occurs or why it matters.
- Give higher scores to reviews that provide specific, evidence-based, traceable, and actionable feedback.
- Be strict but fair.

Please output your evaluation in the following format:

Pertinency: [score from 1 to 10]
Usefulness: [score from 1 to 10]
Evidence-groundedness: [score from 1 to 10]
Traceability: [score from 1 to 10]
Depth of Analysis: [score from 1 to 10]
Comprehensiveness: [score from 1 to 10]
Overall: [score from 1 to 10]

Brief Justification:

[Provide a concise explanation of the scores. Mention the main strengths and weaknesses of the generated review. The justification should focus on review quality, not on the paper's own quality.]

Input:

Paper Content:{paper_content}
Generated Review:{generated_review}

Figure 15: Prompt Template for LLM-as-Judge Evaluation.