Reviewing the Reviewer: Elevating Peer Review Quality through LLM-Guided Feedback

Anonymous ACL submission

Abstract

Peer review is central to scientific quality, yet reliance on simple heuristics—*lazy thinking*—has lowered standards. Prior work treats lazy thinking detection as a single-label task, but review segments may exhibit multiple issues, including broader clarity problems, or *specificity* issues. Turning detection into actionable improvements requires guideline-aware feedback, which is currently missing. We introduce an LLM-driven framework that decomposes reviews into argumentative segments, identifies issues via a neurosymbolic module combining LLM features with traditional classifiers, and generates targeted feedback using issue-specific templates refined by a genetic algorithm. Experiments show our method outperforms zero-shot LLM baselines and improves review quality by up to 92.4%. We also release LAZYREVIEWPLUS, a dataset of 1,309 sentences labeled for *lazy thinking* and *specificity*.

1 Introduction

Peer review is the cornerstone of scientific quality control, ensuring rigorous evaluation of research (Ware and Mabe, 2015). However, the system faces growing strain—particularly in AI, where paper submissions have surged from 1,678 at NeurIPS 2014 to 17,491 in 2024 (10.4× increase) (Wei et al., 2025). This growth, driven by large language models (LLMs) (Liang et al., 2024) and the *publish-or-perish* culture (van Dalen and Henkens, 2012) , has far outpaced the supply of qualified reviewers, even with mandatory reviewing policies.[1] The result is an unsustainable global workload—over 15 million hours annually—that threatens review quality (Aczel et al., 2021).

In NLP research, declining review quality is often attributed to oversimplified heuristics, or *lazy thinking*, which lead reviewers to dismiss

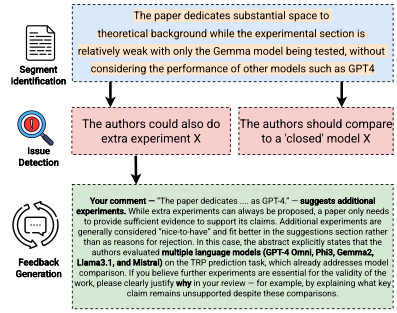


Figure 1: Overall pipeline of our method. We first **identify segments** within a review, then **detect issues**, and finally **generate feedback** to improve each segment.

submissions (Rogers and Augenstein, 2024), accounting for 24.3% author-reported issues in ACL 2023 (Rogers et al., 2023). For example, the **review segment** (an argumentative unit consisting of one or more sentences) in Fig. 1 critiques a paper for using only GPT-4 and requests additional "nice-to-have" experiments. This is an example of the ACL Rolling Review (ARR) guideline issue "authors could also do extra experiment X". [2] Beyond *lazy thinking*, the guidelines also caution against vague phrasing (e.g., *"X is not clear"*) instead of specific feedback (*"X is not clear because of Y"*), referred to as *specificity issues* (Sadallah et al., 2025). The growing need to improve review quality has therefore motivated calls for automated methods that detect both *lazy thinking* and *specificity* issues and deliver corresponding constructive feedback to reviewers (Kuznetsov et al., 2024).

Given the ARR guidelines, issue detection can be formulated as a *multi-label* classification problem by mapping each review segment to a set of issue categories. Naively prompting LLMs yields an F0.5 below 25%, and fine-tuning does not consistently improve performance (Sec. §5), highlighting the generalization challenges due to the abstract nature of issue descriptions and the limited avail-

[1] https://aclrollingreview.org/reviewing-workload-adjustment/

[2] https://aclrollingreview.org/reviewerguidelines

ability of training data.[3] The imbalance between the large input vocabulary and the small annotated training set makes LLMs highly susceptible to overfitting and poor generalization (Bishop, 2006; Rajput et al., 2023; Feng et al., 2023). Recent studies (Vajjala and Shimangaud, 2025; Bucher and Martini, 2024; Han et al., 2025) show that traditional machine-learning models can outperform LLMs on text classification under limited supervision. However, with only a handful of training examples, it remains unclear what are effective representations to mitigate this imbalance.

While Purkayastha et al. (2025) show that lazy thinking annotations can improve review quality, they fall short of providing *constructive verbal feedback*, which is vital for supporting authors in improving their writing (Jansen et al., 2025). In line with this, Thakkar et al. (2025) report improved specificity and professionalism in a large-scale ICLR 2024 trial, though the feedback was not specific to any reviewer guidelines. When prompted to generate feedback, open-weight LLMs often extend reviews ('review extension') or misdirect feedback to authors rather than reviewers ('role separation') (Sec. §3.2), underscoring the need for inference-time methods that produce specific, targeted, and guideline-aware feedback.

Addressing *lazy thinking* and *specificity* is crucial for review quality, but the progress is limited by dataset availability. Prior work (Purkayastha et al., 2025) annotates only a single lazy thinking issue per segment and focuses on "weakness" sections, ignoring specificity and other review sections. However, multiple issue types can co-occur across sections (cf. Figure 1), motivating a new dataset for *multi-label* detection of both issues.

To address these gaps, we present a framework for issue-specific feedback generation. Our framework uses open-weight LLMs to segment reviews, detects issues using a **neuro-symbolic precision-oriented** module that transforms text into structured features through a set of yes/no QA subtasks, classified by an Extra-Trees classifier (Geurts et al., 2006). It then generates actionable feedback for identified issues through *LLM-driven templates* refined with a *genetic algorithm* (Lee et al., 2025) to balance specificity and relevance to the guidelines. Additionally, we introduce LAZYREVIEWPLUS, a dataset of **1,309** expert-annotated review sentences from ARR 2022 and EMNLP 2024 (Dycke et al., 2023), with multi-label annotations for *lazy thinking* and *specificity* issues.

Contributions. Our contributions are: (1) a neuro-symbolic **issue detection** approach that doubles the F0.5 performance of the strongest LLM-only model; (2) an LLM-driven **feedback generation** module that integrates issue-specific templates and a genetic algorithm, improving constructiveness and relevance by about 20% over the best baselines; and (3) controlled review-rewriting studies showing that feedback from our model reduces review issues by up to **92.4**% relative to original reviews; (4) LAZYREVIEWPLUS, the *first* multi-label review dataset annotated with *lazy thinking* and *specificity* for issue identification and feedback generation.

2 LAZYREVIEWPLUS: A single-segment multi-label dataset for *lazy thinking* and *specificity* issue analysis

In this section, we outline the background and the steps involved in constructing our dataset, LAZYREVIEWPLUS, and provide a multi-dimensional analysis of its characteristics.

2.1 Background

In NLP paper reviews, *'lazy thinking'* refers to heuristics used to dismiss papers with limited empirical evidence (Rogers and Augenstein, 2024; Purkayastha et al., 2025). The LAZYREVIEW dataset (Purkayastha et al., 2025) contains **500** segments from ARR 2022 reviews, annotated with **16** lazy thinking classes in the *'weaknesses'* section as a single-segment, multi-class task.[4] Beyond lazy thinking, ARR guidelines warn against unclear or vague phrasing, termed *specificity* issues. The current ARR guidelines outline **18** *lazy thinking* and **7** *specificity* issues in total. In LAZYREVIEWPLUS, we extend this to a single-segment, multi-label setup including both issue types.[5] Our focus is on precision-oriented issue detection and generating verbal feedback to improve review quality, rather than merely signaling *lazy thinking* (Purkayastha et al., 2025).

2.2 Data Procurement and Annotation

Underlying Data. To construct LAZYREVIEWPLUS, we sample 100 reviews (50 each) from ARR

[3] F0.5 is the weighted harmonic mean of the precision and recall.

[4] For a detailed account, see Purkayastha et al. (2025).

[5] All *lazy thinking* issues in Table 9 and *specificity* issues in Table 8 respectively.

2022 and EMNLP 2024 within NLPEER (Dycke et al., 2023), a licensed dataset of ACL and journal reviews (Dycke et al., 2022). Reviews are segmented at the sentence level using SpaCy, covering both *weaknesses* and *comments/suggestions*, unlike LAZYREVIEW which considers only *weaknesses*.[6] This produces 616 and 693 sentences, totaling **1,309** sentences.

Task and Guidelines. We identify review segments with *lazy thinking* and *specificity*. Annotators view the full review along with sentences from the *summary of weaknesses* and *comments, suggestions, and typos* sections. Sentences are labeled using the **B-I-O** scheme (Ramshaw and Marcus, 1999), with each span assigned an issue type. Building on LAZYREVIEW (Purkayastha et al., 2025), we add two categories—*None* and *Not Enough Info*—resulting in **27** issue types (25 from ARR guidelines plus 2 additional classes).[7]

Quality Control. Four Ph.D. students with extensive reviewing experience served as annotators, trained on the finalized guidelines. Disagreements were resolved by a senior student. To ensure reliability, 8% of the dataset (8 reviews; 102 sentences) was double-annotated to compute agreement, while the remaining 92% (92 reviews; 1,207 sentences) was evenly split among annotators. Annotating one review took ∼35 minutes, totaling **180 hours** (∼45 hours per annotator). Inter-Annotator Agreement (Krippendorff's alpha (Krippendorff, 2004)) was **0.78** for **segment identification** (pairwise 0.74–0.82) and **0.53** for **issue detection** (pairwise 0.51–0.55), consistent with prior peer-review studies (Purkayastha et al., 2025) and reflecting the difficulty of identifying *lazy thinking* and *specificity* issues.

2.3 Dataset Analysis

Our dataset, LAZYREVIEWPLUS, contains **1,309** sentences annotated for *lazy thinking* and *specificity* issues (Table 7), comparable to prior human-annotated datasets (Purkayastha et al., 2023, 2025). Segment lengths range from 1–25 sentences (mostly single sentences; Fig. 2a) with 1–8 labels per segment (Fig. 2c), totaling **440** segments. For *lazy thinking*, 'authors should do X' and 'Extra Experiments' dominate (Fig. 2d) which is in line with Purkayastha et al. (2025), while *specificity* issues mostly flag clarity problems ('Unclear X'; Fig. 2b), reflecting common non-constructive feedback patterns (Sadallah et al., 2025).

[6] https://spacy.io/api/sentencizer

[7] Final annotation instructions in Appendix §A.2.

3 Methodology

This section formalizes LLM-driven **feedback generation** and its subtasks: **issue detection** assigns *lazy thinking* and *specificity* issues to segments (Sec. §3.1), and **feedback generation** produces feedback for problematic segments (Sec. §3.2).

3.1 Issue Detection

Task definition and challenges. For **issue detection**, For a predefined label set of issues, $\mathcal{L} = \{\ell_1, \ldots, \ell_m\}$, the task is to each segment $S = (s_a, \ldots, s_b)$ in a review. Formally, an issue classification function $g(S) \subseteq \mathcal{L}$ maps each segment to a subset of labels, yielding $G(R) = (g(S_1), \ldots, g(S_k))$ for a segmented review R.

Following prior work on issue identification (Purkayastha et al., 2025), we first conducted zero-shot classification using open-source LLMs. We found that even the best-performing LLM fails to identify at least **56%** of the frequently occurring issues in our dataset, with or without fine-tuning.[8] Issue detection is fundamentally a reasoning problem, as it often requires intermediate reasoning steps to determine issue labels rather than relying solely on surface-level similarity between review segments. For example, justifying whether a requested experiment is *nice-to-have* or *must-have* is a necessary intermediate step before assigning the corresponding issue label. However, because the input vocabulary size is several orders of magnitude larger than the number of annotated review segments, it is infeasible to rely on mainstream approaches, such as fine-tuning, to endow LLMs with this specialized reasoning capability.

Approach. To address those challenges, we propose a **neuro-symbolic** method to decompose the issue detection task into two stages: *(1) Abstract Feature Extraction* and *(2) Precision-oriented Machine Learning-based Classification*.

1. Abstract Feature Extraction We prompt LLMs to answer a fixed set of Yes/No questions for each issue and encode their responses into fixed-length structured feature vectors, whose dimensionality is comparable to the number of training instances. This design transforms the original QA-style reasoning task into a pattern recognition problem in a compact, structured feature space and abstracts

[8] See details in Figure 6

away review-specific details. More specifically, for each issue $l \in \mathcal{L}$, an LLM is first prompted to generate an inventory of issue-specific questions $Q_i^l = \{q_1, q_2, \ldots, q_{F_c}\}$, by comparing and finding patterns between review segment $s_i \in R_l$ (Review set annotated as class l), where $F_c = 10$ is the number of questions per issue. Consequently, we employ the same LLM to answers these questions as Yes/No/Not relevant based on each segment. The resulting set of answers for all issues, $A_i = \{A_i^l \mid l \in \mathcal{L}\}$, is converted into a feature vector $\mathbf{v}_i \in \{-1, 0, 1\}^{|\mathcal{L}| \times F_c}$, where each entry encodes the LLM's judgment: 1 for Yes, -1 for No, and 0 for Not relevant. These feature vectors thus provide discriminative and abstract signals for identifying the different issue types.

2. Precision-Oriented Classification Training classical machine learning models on those structured feature vectors is necessary because the extracted features are correlated but not deterministically linked to issue labels. Furthermore, to build trust among reviewers by minimizing false alarms, we prioritize precision over recall in issue detection. Accordingly, we optimize for the precision-oriented $\mathbf{F_{0.5}}$ metric by selecting models maximizing $F_{0.5}$ on the validation set.

Compared to existing approaches that incorporate LLMs into traditional machine learning pipelines (Manikandan et al., 2023; Jeong et al., 2025), our method is the *first* to transform raw text inputs into abstract structured representations that explicitly encode intermediate reasoning results. This approach also provides greater interpretability than end-to-end LLM-based methods.

3.2 Feedback Generation

Task definition and challenges. Let each segment be s_i with label l_i and optional context C (e.g., abstract A). The goal is to produce feedback $F_i = g(s_i, l_i \mid C)$; without context, $F_i = g(s_i, l_i)$. For a review $R = (s_1, \ldots, s_n)$ with labels $L = (l_1, \ldots, l_n)$, the feedback sequence is $F = (F_1, \ldots, F_n)$, supporting segment-only and context-informed generation.

Building on prior work on **feedback generation** (Thakkar et al., 2025), we first explored zero-shot generation using open-source LLMs. We found that most outputs either produced full review rewrites ("review extension") or mistakenly targeted authors instead of reviewers ("role separation"). In our LAZYREVIEWPLUS dataset, at least **41%** of outputs from leading open-source LLMs were review extensions, while at least **23%** exhibited role separation. These observations highlight the need for an inference-time feedback generation approach that explicitly addresses these issues.[9]

Approach. To ensure targeted and diverse reviewer feedback, we adopt a Genetic Evolution-inspired algorithm aimed at diversifying responses across the detected issues (Lee et al., 2025). Each piece of feedback is tailored to its corresponding issue, making it more actionable and constructive. The algorithm operates in six stages: *(1) Template Construction*, *(2) Plan Generation*, *(3) Population Initialization*, *(4) Fitness Evaluation*, *(5) Parent Selection and Crossover*, and *(6) Final Candidate Selection*. Stages (4)-(6) are repeated till the maximum number of iterations are reached within the genetic algorithm.

1. Template Construction LLMs often produce less diverse or hallucinated outputs when unconstrained. Templates offer scaffolds that reduce hallucinations while maintaining control over diversity, effectively mitigating *review extension* (Kang et al., 2025; Xu et al., 2024). We therefore design **25** author-crafted issue-specific templates $T = \{t_1, \ldots, t_{25}\}$, each aligned with ACL ARR guideline issue types (Rogers and Augenstein, 2024).[10] All templates are verified and refined by senior authors for consistency and fit.

2. Plan Generation To promote diversity in generated feedback, prior work uses model-generated plans to guide unconstrained generation and reduce uncertainty (Narayan et al., 2023; Huot et al., 2023). For each review segment s_i and issue type l_i, we prompt the LLM to produce a plan enriching the template using the paper's abstract, reviewer summary, and noted strengths—mitigating the *role separation* bias. The planner selects relevant knowledge k_i and justifies it with an explanation e_i, forming the plan $\mathcal{P}(r_i, l_i) = \{(k_i, e_i) \mid i = 1, \ldots, N\}$.

3. Population Initialization The LLM is then prompted with the review segment, s_i, the plan, P and the template, t_i to generate n sets of candidates. The initial population set, M_0 is thus given as, $F_0 = \{fb_1, fb_2, \ldots, fb_n\}$.

4. Fitness Evaluation Since we lack ground-truth feedback, we employ composite, verifiable rewards that combine positive scores and penalties to balance each generated response (Hu et al., 2020). We define a fitness function based solely on in-

[9] Experimental details and examples are in §A.6.2

[10] Full list in Table 10

trinsic text properties. Let n_{sent} and n_{words} denote the number of sentences and words in the feedback. Building on prior work (Yaacoub et al., 2025) that suggests automated feedback should be concise and readable, we quantify such measures using automated metrics. To encourage conciseness, we reward shorter feedback: $sc_{len} = \frac{\min(n_{\text{sent}}, n_{\max})}{n_{\max}}$. Template adherence ensures on-topic responses, measured via n-gram overlap: $sc_{temp} = \frac{|\text{n-grams}_{\text{fb}} \cap \text{n-grams}_{\text{temp}}|}{|\text{n-grams}_{\text{temp}}|}$. Readability is encouraged using the Flesch Reading score (Flesch, 1948): $sc_{read} = \frac{\text{Flesch score}}{100}$. To maintain professional and targeted feedback, greetings or any off-task expressions are penalized: $pen_{forb} = \frac{\#\text{forbidden terms}}{n_{\text{words}}}$. The overall fitness is a normalized sum of these components: $fit(F_0) = sc_{len} + sc_{temp} + sc_{read} - pen_{forb}$. This formulation allows evaluation independently of external context, making it suitable for self-contained optimization in LLM-driven **feedback generation**.

5. Parent Selection and Crossover To **select parents** for evolving the population, we employ Boltzmann Tournament Selection (Goldberg, 1990). Fitness scores are converted into probabilities via a softmax function: $P(fb_i) = \frac{\exp(fit(fb_i)/\tau)}{\sum_{j=1}^{m} \exp(fit(fb_j)/\tau)}$, where τ is a temperature parameter controlling the sharpness of the selection distribution. Candidates are sampled according to $P(fb_i)$, allowing high-quality feedback to be favored while maintaining diversity. Exactly $n_{parents}$ are selected for the next crossover phase. The **crossover function** takes parent feedback candidates, e.g., fb_1 and fb_2, and generates new offspring: $Cross(fb_1, fb_2) \rightarrow$ new candidates. In our approach, the LLM is prompted to produce offspring that integrate features from both parents, generating novel and diverse feedback candidates.

6. Final Candidate selection After a fixed number of generations,n_{gen}, we select the candidate with the maximal fitness score, $fb^* = \arg\max_{fb_i \in \mathcal{M}} fit(fb_i)$, where $\mathcal{M}$ is the set of final candidates. If multiple candidates achieve the same maximal score, we perform a cross-over of these candidates to generate a new candidate, which is taken as the final solution.

4 Experimental Details

Models. We evaluate the robustness of our approach using LLMs from diverse model families for all the tasks within our pipeline. Specifically, we employ Qwen 2.5 7B Instruct (Yang et al., 2025) (Qwen), Yi 1.5 9B Chat (Young et al., 2025) (Yi), Deepseek LLM 7B Chat (Bi et al., 2024) (Deep.), Phi-4 14B (Abdin et al., 2024) (Phi), and GPT OSS 20B (Agarwal et al., 2025) (Oss.). For **issue detection**, we employ various machine learning classifiers—K Nearest Neighbour (`KNN`), Logistic Regression (L1/L2; `LogReg-L1/LogReg-L2`), Random Forest (`RF`), Decision Trees (`DT`), Support Vector Machines (Linear/RBF/Polynomial; `SVM-Lin/SVM-RBF`), Gradient Boost (`GBoost`), AdaBoost (`AdaB`), Extra Trees (`ExtraT`), and Multilayer Perceptron (`MLP`).

Evaluation Metrics. For **issue detection**, we report Precision, Recall and F0.5 to prioritize high precision. For **feedback generation**, lacking ground-truth references, we conduct both **automated** and **human** evaluation. For **automated evaluation**, we use `Prometheus-V2` (Kim et al., 2024) to score Constructiveness, Conciseness, Relevance, and Specificity on a 1–5 scale, following prior work (Sahnan et al., 2025; Maurya et al., 2025) since it outperforms Prometheus-V1 and GPT-4o (Hurst et al., 2024) (cf. §A.6.1). For **human evaluation**, following prior work in peer reviewing (Purkayastha et al., 2025; Lu et al., 2025), we recruited three Ph.D. students fluent in English and experienced in peer review to rate 100 responses per issue type for the best-performing model, using the same criteria to assess the practical quality and usefulness of the feedback.

Data Split and Validation. Since our proposed **issue detection** approach involves training multiple classifiers, we use 90% of the data for 5-fold cross-validation to evaluate model performance, and 10% for hyperparameter tuning by splitting at the review level. We develop a distance-sensitive algorithm to partition the dataset in a way that reflects real-world prevalence, where each review belongs exclusively to a fold in the cross-validation set up.[11] However, **feedback generation** does not require training, so results are reported on the full dataset.

Baselines. For a review segment s_i and the issue set $\mathcal{L} = l_1, \ldots, l_n$, we evaluate the following baselines. For **issue detection**: (i) Zero-shot ($\texttt{LLM}_{\texttt{Zero}}$): the LLM predicts $l_i \subseteq \mathcal{L}$ given s_i; (ii) Finetuning ($\texttt{LLM}_{\texttt{Fine}}$): we finetune the LLM on the training split and evaluate with cross-validation; (iii) QA-based classification ($\texttt{LLM}_{\texttt{QA}}$ / $\texttt{LLM}_{\texttt{QAFine}}$): the pretrained LLM ($\texttt{LLM}_{\texttt{QA}}$) or its finetuned variant ($\texttt{LLM}_{\texttt{QAFine}}$) classifies s_i using the same

[11] Example in Figure 3

Method	Yi	Phi	Qwen	Deep.	Oss.
$\texttt{LLM}_{\texttt{Zero}}$	0.04	0.05	0.06	0.03	0.23
$\texttt{LLM}_{\texttt{Fine}}$	0.23	0.12	0.13	0.11	0.19
$\texttt{LLM}_{\texttt{QA}}$	0.05	0.12	0.17	0.20	0.12
$\texttt{LLM}_{\texttt{QAFine}}$	0.10	0.10	0.07	0.08	0.11
RoBERTa			0.39		
KNN	0.34	0.38	0.39	0.38	0.37
LogReg-L2	0.32	0.37	0.31	0.31	0.33
LogReg-L1	0.32	0.38	0.35	0.32	0.32
RF	0.34	0.41	0.40	0.35	0.33
DT	0.26	0.27	0.24	0.22	0.25
SVM-RBF	0.34	0.35	0.35	0.34	0.32
SVM-Lin	0.41	0.42	0.38	0.35	0.34
GBoost	0.46	0.49	0.43	0.36	0.41
AdaB	0.43	0.48	0.36	0.36	0.37
ExtraT	**0.51**	**0.51**	**0.44**	**0.42**	**0.44**
MLP	0.45	0.49	0.41	0.42	0.41

Table 1: Cross-Validation Performance comparison of methods for the **issue detection** task across LLMs and the encoder-only baseline (RoBERTa) based on F0.5 scores. The best performing classifier is in bold with the best performing LLM highlighted.

feature-specific questions and answers as in our approach. (iv) Encoder-Only Baseline: We use RoBERTa (Liu et al., 2019) as a baseline, trained on the same data as the $\texttt{LLM}_{\texttt{Fine}}$ model, to evaluate the impact of task-specific finetuning using smaller models. For **feedback generation**: (i) 1-pass (1-pass): the LLM generates feedback from s_i and l_i; (ii) Template-guided (Temp): the LLM incorporates a template t_i with s_i and l_i; (iii) Plan-then-generate (Plan): the LLM devises a plan $\mathcal{P}i$ for l_i and then integrates it into t_i to generate feedback; (iv) Best-of-N (BoN) (Brown et al., 2024): we sample $N = n \times n_{gen}$ candidates and select the best using our fitness function; (v) Self-refinement (Self-Ref.) (Madaan et al., 2023): we sample n candidates and refine each over n_{gen} iterations, retaining the final output.[12]

5 Results and Discussion

In this section, we present the results of different subtasks within our approach. Following prior work (Lan et al., 2024; Pichler et al., 2025), for **segment identification**, we adapt a zero-shot approach of detecting segments within a review as tagging sentences with a B/I/O tag. We achieve an Precision, Recall and F1 score of **0.81**, **0.77** and **0.79** respectively using Phi.[13] The results for the rest of the approaches are outlined below.

1. Precision of the issue detection approach

Overall results. We report overall performances in Table 1. We find that our neuro-symbolic approach, which combines LLM-extracted features with a classical ML classifier, achieves the best results across the board, surpassing zero-shot performance by at least **0.9** points. However, our encoding-only baseline, RoBERTa outperforms the other LLM-based baselines reaffirming the fact that task-specific finetuning with less data can be achieved using smaller models more effectively than that of LLMs. Fine-tuning LLMs on instances from our dataset also improves performance significantly (0.04–0.23 in terms of Yi), highlighting the need for a specialized dataset; however, the gains still fall short of our neuro-symbolic approach, likely due to the high data requirements for task-specific LLM adaptation. Using feature-aligned questions in $\texttt{LLM}_{\texttt{QA}}$ improves performance over zero-shot variants, confirming the value of the Q/A module. The $\texttt{LLM}_{\texttt{QAFine}}$ variant, however, underperforms $\texttt{LLM}_{\texttt{QA}}$, likely due to data-constrained segment-level finetuning. Our results show that Extra Trees with Phi features delivers the strongest precision-oriented **issue detection**, benefiting from structured neuro-symbolic representations and the ability of tree-based models to capture non-linear feature interactions. The low variance (below 0.01) further confirms the robustness and diversity of our dataset (cf. Table 19).[14].

Ablation Study. We conduct a series of ablation experiments to evaluate the effectiveness of our approach: **1. Representation Effectiveness:** We employed a sentence embedding model to transform review segments into vector embeddings and evaluated their classification performance. Overall, the results obtained for the best performing classifier are around **0.5** points worse than those achieved with our feature-vector-based approach.[15] **2. Feature question size**: We evaluate the impact of reducing the feature vector size from 10 to 5 using our best-performing model, Phi, due to its superior feature extraction capabilities. We find that reducing the feature vector leads to a **0.12** point drop in F0.5 score in cross-validation[16]. This indicates that the smaller feature set fails to capture sufficient information, likely causing underfitting and limiting the model's representational capacity. **3. Effect of various thresholds**: We evaluate our approach against the strongest baseline, RoBERTa, across thresholds $0.25, 0.5, 0.75, 1.0, 2.0$ (Table 17). At $\beta = 0.25$, Extra Trees with Phi features achieves high preci-

[12] Hyperparameters are listed in Table 24.

[13] Details about the experiment are in §A.4.

[14] Results with human-written questions in Sec. §A.5.5

[15] We employ the widely used all-MiniLM-L6-v2 sentence transformer model for this experiment. Full results are provided in Table 14.

[16] Details in Table 15.

Metric	Model	1-pass	BoN	Self-Ref.	Temp	Plan	Ours
Const.	Yi	1.9	2.0	2.3	2.8	3.0	3.9
Rel.	Yi	2.0	2.2	2.4	2.9	3.1	3.8
Const.	Phi	2.1	2.2	2.4	3.1	3.3	**4.3**
Rel.	Phi	2.2	2.3	2.6	3.2	3.4	**4.3**
Const.	Qwen	1.8	2.0	2.2	2.9	3.1	3.8
Rel.	Qwen	1.9	2.1	2.4	3.0	3.2	3.8
Const.	Deep.	1.7	1.8	2.4	2.7	2.9	3.5
Rel.	Deep.	1.8	1.9	2.4	2.8	3.0	3.6
Const.	Oss.	1.9	2.0	3.0	2.8	3.0	4.0
Rel.	Oss.	2.0	2.1	3.1	2.9	3.1	4.1

Table 2: Model performance on **feedback generation** across six methods for **Constructiveness (Const.)** and **Relevance (Rel.)**. Overall best results are **bolded**, method-wise best are underlined.

sion (0.90) but low recall (0.24), favoring conservative detections. Increasing β (0.5–2.0) boosts recall at the cost of precision (e.g., 0.42/0.67 at $\beta = 2.0$), illustrating tunable precision–recall trade-offs. As shown in Fig. 8, our approach consistently outperforms RoBERTa across all thresholds, providing reliable signals while maintaining a balanced precision–recall trade-off and reducing false alarms.

2. Effectiveness of feedback generation

Overall Results. We present the results of our **automated evaluation** in Table 2.[17] We find that our approach consistently outperforms all baselines, producing targeted, diverse feedback with the same number of candidates. While Best-of-N (BoN) boosts diversity, it lacks structure; Self-refinement (Self-refine) improves quality but still drifts off-task without external rewards (Lee et al., 2025).We also find that template-based prompting (Temp) boosts performance—Phi improves from 2.1 to 3.1 in *Constructiveness* and 2.2 to 3.2 in *Relevance* (Xu et al., 2024; Kang et al., 2025). The Plan-then-Generate paradigm yields modest gains (e.g., *Constructiveness* 3.1 $\rightarrow$ 3.3), suggesting that structured planning helps, though template scaffolding drives most of the improvement. Overall, we find Phi to be the strongest performer, likely due to its scale and diverse pretraining (Abdin et al., 2024). For **human evaluation**, we achieve substantial inter-annotator agreement in terms of Krippendorff's α across all metrics (*Constructiveness*: 0.65, *Relevance*: 0.68, *Specificity*: 0.72, *Conciseness*: 0.72). All agreement values are statistically significant: Constructiveness CI = [0.603, 0.697], Relevance CI = [0.635, 0.725], and both Specificity and Conciseness CI = [0.678, 0.762], indicating the results are robust and likely generalizable to a larger population. The results for the **human evaluation** (cf. Table 25) for the best-performing model, Phi reveals a similar pattern as our **automated evaluation**, with our approach outperforming baselines across metrics and the rankings across baselines being the same. The human alignment with the evaluator, Prometheus is strong (Spearman ρ = 0.85), validating the automated metrics used.[18]

[17] Full results in Table 13

Variant	Const.	Rel.
Full Algorithm (Ours)	**4.3**	**4.3**
w/o. Template Construction	3.6	3.5
w/o. Plan Generation	3.8	3.8
w/o. Population Initialization	3.9	3.9
w/o. Fitness Evaluation	3.7	3.7
w/o. Parent Selection & Crossover	3.9	3.8
w/o. Final Candidate Selection	3.8	3.7

Table 3: Ablation study of various components for the best performing model, Phi within the genetic algorithm focusing on the automated metrics.

Ablation Study. We perform a series of ablation studies to establish the contribution of each aspect within our approach. **i. Effect of components within our approach**: We ablate our algorithm in Table 3 for Phi and find that removing the Template Construction phase drops Constructiveness by $\sim$ 0.7 points for Phi, as templates provide structured scaffolds that reduce *review extension*. Removing the fitness function lowers scores by $\sim$ 0.6 points and final candidate is often longer than its parents, revealing *length bias*, a common reward hacking shortcoming in LLM outputs (Emberson et al., 2025). We obtain similar results for the other models in this work.[19] **ii. Effect of individual rewards**: We analyze each reward in our *fitness function* (cf. Table 28) and find that removing the template adherence (sc_{temp}) reward drops Constructiveness by 0.6 points for Phi as outputs go off-topic. Removing the length-based reward (sc_{len}) reduces Conciseness by 0.6 points, reflecting how longer responses become convoluted and harder to read. Removing the cross-over component also leads to a decline in the constructiveness scores by 0.4 points pointing to the selection of less effective candidates. **iii. Effect of additional information in the Planner**: We show the ablation results in Table 27. Our full approach uses Abstract, reviewer-written summary, and reviewer-written strengths. Removing a single component reduces scores by 0.1–0.3 points (constructiveness for Phi), and ablating multiple components lowers performance further (up to 0.6 points) (constructiveness), demonstrating that each input contributes complementary guidance. This is especially important for frequent issues like "ex-

[18] The alignment study is in §A.6.1

[19] Full ablation for all models in Table 26.

tra experiments," which require all components to produce targeted, actionable feedback.

6 Expert Assessment of Review Quality Improvement Through Feedback

Setup. To evaluate the quality of the generated feedback, we conduct a controlled study comparing the original reviews, rewrites based only on the detected *lazy thinking* and *specificity issues*, and rewrites that additionally incorporate the generated feedback. Following Purkayastha et al. (2025), we form two groups of two Ph.D. students each. One rewrites 50 reviews using only issues from our **issue detection** module emulating the setup in Purkayastha et al. (2025), while the other uses both issues and targeted feedback from our **feedback generation** approach, marking feedback as *actionable* or not. Rewrites are evaluated on **Constructiveness**, **Justified**, and **Adherence**. Two senior students perform pairwise comparisons between original reviews, issue-only rewrites, and issue-plus-feedback rewrites, splitting 50 reviews evenly and reserving 10 for agreement.

Results. We report the win-tie-loss results for the controlled experiment in Table 4. Reviews rewritten with feedback (Issue det. *w/* feed) outperform both reviews written with only **issue detection** (Issue det.) and the original reviews (Orig), achieving 85% and 95% higher Constructiveness, respectively. Notably, even feedback based solely on detected issues significantly improves over the original reviews, consistent with Purkayastha et al. (2025). Reviewers in the feedback-based rewrite group incorporated 96% of the feedback, marking it actionable, indicating high-quality guidance. Feedback-based rewrites reduce identified issues in reviews up to **92.4**%, compared to 85.2% for rewrites with only **issue detection**, demonstrating the effectiveness of the **feedback generation** module in driving review quality. In line with Purkayastha et al. (2025), a Bradley-Terry preference ranking model (Bradley and Terry, 1952) trained on adherence preferences yields scores of 1.2 for feedback-based rewrites, -1.0 for original reviews, and 0.6 for **issue detection**-based rewrites. This corresponds to a 93.4% win rate over original texts and 78.5% over issue-detection rewrites, further supporting the effectiveness of the annotations. Inter-annotator agreement (Krippendorff's α) is 0.71, 0.73, and 0.77 for Constructiveness, Justification, and Adherence, respectively.

Type	**Cons.** (W/T/L)	**Just.** (W/T/L)	**Adh.** (W/T/L)	**Δ Issue** (↓)
Issue det. vs. Orig	85/5/10	80/5/15	85/5/10	85.2
Issue det. *w* feed vs Orig	95/5/0	95/5/0	90/5/5	**92.4**
Issue det. *w* feed vs. Issu det.	85/10/5	90/5/5	85/10/5	78.5

Table 4: Comparison of Original, Issue-detected, and Issue-detected with feedback re-written reviews on Constructiveness, Justification, and Adherence using Win/Tie/Loss rates. **ΔIssue** indicates issue reduction.

7 Related Work

Review Quality Assessment. As scientific publications rise, reviewers face increasing burden, driving interest in assessing review quality (Kuznetsov et al., 2024). Prior studies have measured tonality (Bharti et al., 2024), thoroughness (Lu et al., 2025). . We focus on detecting and providing feedback for review comments that exhibit cognitive biases (*lazy thinking*) and stylistic issues, following ACL ARR guidelines (Rogers and Augenstein, 2024). Sadallah et al. (2025) emphasizes evaluating reviews based on their constructiveness for author revisions. In contrast, we focus on precisely identifying cognitive biases and providing targeted, diverse feedback to reviewers to improve overall review quality.

LLM-generated feedback. LLMs have been widely used for feedback, especially in education, supporting English learners (Kasneci et al., 2023; Escalante et al., 2023), coding instruction (Misiejuk et al., 2024; Azaiz et al., 2024), and TOEFL preparation (Kasneci et al., 2023). Studies show LLM feedback matches human quality and drives positive outcomes. Building on this, Thakkar et al. (2025) used Claude at ICLR 2025 to improve review specificity, correctness, and politeness. However, their method ignores guideline compliance and relies on a closed-source model. We address both by enforcing ACL ARR adherence and using open-source LLMs, ensuring integrity and privacy.

8 Conclusion

To enhance review quality, we introduce an open-source, LLM-driven framework that identifies issues precisely and generates targeted feedback via iteratively refined templates. It outperforms zero-shot and fine-tuned LLM-based baselines, and explicit feedback further improves review quality. Additionally, we release LAZYREVIEWPLUS, an expert-annotated dataset of review sentences labeled for *lazy thinking* and *specificity*. We hope our work inspires further efforts to enhance the quality of peer review.

Limitations

In this work, we propose a new dataset alongside a precision-oriented issue identification and feedback generation framework designed to enhance the quality of peer reviews. However, our approach comes with several limitations.

First, we introduce LAZYREVIEWPLUS, a single-segment, multi-label classification dataset for detecting *lazy thinking* and *specificity* issues in peer reviews. The dataset currently covers only conferences hosted on ACL Rolling Review (ARR), due to their well-defined guidelines for *lazy thinking* and *specificity*. Extending this resource to other domains, conferences, or languages would require developing equally rigorous guidelines with precise definitions for these issues.

Second, in terms of methodology, we conduct extensive experiments and analyses for both **issue identification** and **feedback generation**. However, these results may not be fully generalizable to other scientific or evaluative tasks, and thus the results should be interpreted with caution. Moreover, our feedback generation approach relies on predefined templates that are refined using a genetic algorithm. Over time, however, reviewers may become desensitized to recurring feedback styles. The long-term effectiveness and robustness of a template-driven approach therefore warrant further investigation, especially in repeated or large-scale deployment settings.

Third, our framework relies on large language models, which may carry stylistic or cultural biases inherited from their training data. Although peer reviewing is intended to be domain-neutral, such biases may still manifest subtly in phrasing or prioritization. Future work could incorporate bias detection and mitigation strategies to enhance fairness and neutrality in feedback generation.

Finally, while automated feedback systems can assist reviewers and improve the clarity and constructiveness of reviews, they should complement—rather than replace—human judgment. The ultimate responsibility for fair, constructive, and contextually informed reviewing must remain with human reviewers.

Ethics Statement

Before commencing the project, we obtained the necessary ethics approvals from the Ethics Committees of the leading institute. Our dataset, LAZYREVIEWPLUS, will be released under the CC-BY-NC 4.0 license. The underlying ARR reviews were collected from NLPEER (Dycke et al., 2023), which is licensed under the same terms. The analysis and automatic annotation processes do not involve any personal or sensitive information. The annotators participating in the user study and dataset annotation were compensated at a rate of $25 per hour, ensuring fair remuneration.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, et al. 2024. Phi-4 technical report. *ArXiv preprint*.

ACL Publication Ethics Committee. 2024. ACL Policy on Publication Ethics. *Association for Computational Linguistics*.

Balazs Aczel, Barnabas Szaszi, and Alex O Holcombe. 2021. A billion-dollar donation: estimating the cost of researchers' time spent on peer review. *Research integrity and peer review*, 6(1):1–8.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, et al. 2025. gpt-oss-120b & gpt-oss-20b model card. *Arxiv preprint*.

Imen Azaiz, Natalie Kiesler, and Sven Strickroth. 2024. Feedback-Generation for Programming Exercises with GPT-4. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, ITiCSE 2024, page 31–37, New York, NY, USA. Association for Computing Machinery.

Prabhat Kumar Bharti, Meith Navlakha, Mayank Agarwal, and Asif Ekbal. 2024. PolitePEER: does peer review hurt? A dataset to gauge politeness intensity in the peer reviews. *Language Resources and Evaluation*, 58(4):1291–1313.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, et al. 2024. DeepSeek LLM: Scaling Open-Source Language Models with Longtermism. *Arxiv preprint*.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*, volume 4. Springer-Verlag, Berlin, Heidelberg.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *Arxiv preprint*.

Martin Juan José Bucher and Marco Martini. 2024. Fine-Tuned 'Small' LLMs (Still) Significantly Outperform Zero-Shot Generative AI Models in Text Classification. *ArXiv preprint*.

Nils Dycke, Ilia Kuznetsov, and Iryna Gurevych. 2022. Yes-yes-yes: Proactive data collection for ACL rolling review and beyond. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 300–318, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nils Dycke, Ilia Kuznetsov, and Iryna Gurevych. 2023. NLPeer: A unified resource for the computational study of peer review. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5049–5073, Toronto, Canada. Association for Computational Linguistics.

Luke Emberson, Ben Cottier, Josh You, Tom Adamczewski, and Jean-Stanislas Denain. 2025. LLM responses to benchmark questions are getting longer over time. *Blog Post*.

Juan Escalante, Austin Pack, and Alex Barrett. 2023. AI-generated feedback on writing: Insights into efficacy and ENL student preference. *International Journal of Educational Technology in Higher Education*, 20(1):57.

Tao Feng, Lizhen Qu, and Gholamreza Haffari. 2023. Less is more: Mitigate spurious correlations for open-domain dialogue response generation models by causal discovery. *Transactions of the Association for Computational Linguistics*, 11:511–530.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine learning*, 63(1):3–42.

David E Goldberg. 1990. A note on boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing. *Complex Systems*, 4:445–460.

Xu Han, Yumeng Sun, Weiqiang Huang, Hongye Zheng, and Junliang Du. 2025. Towards Robust Few-Shot Text Classification Using Transformer Architectures and Dual Loss Strategies. *Arxiv preprint*.

Junjie Hu, Yu Cheng, Zhe Gan, Jingjing Liu, Jianfeng Gao, and Graham Neubig. 2020. What Makes A Good Story? Designing Composite Rewards for Visual Storytelling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7969–7976.

Fantine Huot, Joshua Maynez, Shashi Narayan, Reinald Kim Amplayo, Kuzman Ganchev, Annie Priyadarshini Louis, Anders Sandholm, Dipanjan Das, and Mirella Lapata. 2023. Text-blueprint: An interactive platform for plan-based conditional generation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 105–116, Dubrovnik, Croatia. Association for Computational Linguistics.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, et al. 2024. GPT-4o System Card. *Arxiv preprint*.

Thorben Jansen, Lars Höft, J Luca Bahr, Livia Kuklick, and Jennifer Meyer. 2025. Constructive feedback can function as a reward: Students' emotional profiles in reaction to feedback perception mediate associations with task interest. *Learning and Instruction*, 95:102030.

Daniel P Jeong, Zachary Chase Lipton, and Pradeep Kumar Ravikumar. 2025. LLM-select: Feature selection with large language models. *Transactions on Machine Learning Research*, 2025.

Xiaoqiang Kang, Zimu Wang, Xiaobo Jin, Wei Wang, Kaizhu Huang, and Qiufeng Wang. 2025. Template-Driven LLM-Paraphrased Framework for Tabular Math Word Problem Generation. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence, February 29-March 4, 2025, Philadelphia, Pennsylvania, USA*, pages 24303–24311. AAAI Press.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

Klaus Krippendorff. 2004. Reliability in content analysis: Some common misconceptions and recommendations. *Human communication research*, 30(3):411–433.

Ilia Kuznetsov, Osama Mohammed Afzal, Koen Dercksen, Nils Dycke, Alexander Goldberg, Tom Hope, Dirk Hovy, Jonathan K. Kummerfeld, Anne Lauscher, et al. 2024. What Can Natural Language Processing Do for Peer Review? *Arxiv preprint*.

Mengfei Lan, Lecheng Zheng, Shufan Ming, and Halil Kilicoglu. 2024. Multi-label sequential sentence classification via large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16086–16104, Miami, Florida, USA. Association for Computational Linguistics.

Kuang-Huei Lee, Ian Fischer, Yueh-Hua Wu, Dave Marwood, Shumeet Baluja, Dale Schuurmans, and Xinyun Chen. 2025. Evolving Deeper LLM Thinking. *Arxiv Preprint*.

Weixin Liang, Yaohui Zhang, Zhengxuan Wu, Haley Lepp, Wenlong Ji, Xuandong Zhao, Hancheng Cao, Sheng Liu, Siyu He, Zhi Huang, Diyi Yang, Christopher Potts, Christopher D Manning, and James Y. Zou. 2024. Mapping the increasing use of LLMs in scientific papers. In *First Conference on Language Modeling, University of Pennsylvania, Philadelphia, PA, October 7-9, 2024*. Openreview.net.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *Arxiv preprint*.

Sheng Lu, Ilia Kuznetsov, and Iryna Gurevych. 2025. Identifying aspects in peer reviews. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6145–6167, Suzhou, China. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, et al. 2023. Self-Refine: Iterative Refinement with Self-Feedback. In *Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems, NeurIPS 2023, New Orleans, Louisiana, USA, Dec 10-16, 2023*. Openreview.net.

Hariharan Manikandan, Yiding Jiang, and J Zico Kolter. 2023. Language models are weak learners. In *Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems, NeurIPS 2023, New Orleans, Louisiana, USA, Dec 10-16, 2023*. Openreview.net.

Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025. Unifying AI tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of LLM-powered AI tutors. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1234–1251, Albuquerque, New Mexico. Association for Computational Linguistics.

Kamila Misiejuk, Rogers Kaliisa, and Jennifer Scianna. 2024. Augmenting assessment with AI coding of online student discourse: A question of reliability. *Computers and Education: Artificial Intelligence*, 6:100216.

Shashi Narayan, Joshua Maynez, Reinald Kim Amplayo, Kuzman Ganchev, Annie Louis, Fantine Huot, Anders Sandholm, Dipanjan Das, and Mirella Lapata. 2023. Conditional generation with a question-answering blueprint. *Transactions of the Association for Computational Linguistics*, 11:974–996.

Viet Thanh Pham, Lizhen Qu, Zhuang Li, Suraj Sharma, and Gholamreza Haffari. 2025. SurveyPilot: an agentic framework for automated human opinion collection from social media. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4397–4422, Vienna, Austria. Association for Computational Linguistics.

Axel Pichler, Janis Pagel, and Nils Reiter. 2025. Evaluating LLM-prompting for sequence labeling tasks in computational literary studies. In *Proceedings of the 9th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2025)*, pages 32–46, Albuquerque, New Mexico. Association for Computational Linguistics.

Sukannya Purkayastha, Anne Lauscher, and Iryna Gurevych. 2023. Exploring jiu-jitsu argumentation for writing peer review rebuttals. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14479–14495, Singapore. Association for Computational Linguistics.

Sukannya Purkayastha, Zhuang Li, Anne Lauscher, Lizhen Qu, and Iryna Gurevych. 2025. LazyReview: A dataset for uncovering lazy thinking in NLP peer reviews. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3280–3308, Vienna, Austria. Association for Computational Linguistics.

Daniyal Rajput, Wei-Jen Wang, and Chun-Chuan Chen. 2023. Evaluation of a decided sample size in machine learning applications. *BMC bioinformatics*, 24(1):48.

L. A. Ramshaw and M. P. Marcus. 1999. *Text Chunking Using Transformation-Based Learning*, pages 157–176. Springer Netherlands, Dordrecht.

Anna Rogers, Marzena Karpinska, Jordan Boyd-Graber, and Naoaki Okazaki. 2023. Program chairs' report on peer review at acl 2023. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages xl–lxxv, Toronto, Canada. Association for Computational Linguistics.

Anne Rogers and Isabelle Augenstein. 2024. How to review for ACL Rolling Review. *ACL Rolling Review*.

Abdelrahman Sadallah, Tim Baumgärtner, Iryna Gurevych, and Ted Briscoe. 2025. The good, the bad and the constructive: Automatically measuring peer review's utility for authors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28979–29009, Suzhou, China. Association for Computational Linguistics.

Dhruv Sahnan, David Corney, Irene Larraz, Giovanni Zagni, Ruben Miguez, Zhuohan Xie, Iryna Gurevych, Elizabeth Churchill, Tanmoy Chakraborty, and Preslav Nakov. 2025. Can LLMs Automate Fact-Checking Article Writing? *Arxiv preprint*.

Nitya Thakkar, Mert Yuksekgonul, Jake Silberg, Animesh Garg, Nanyun Peng, Fei Sha, Rose Yu, Carl Vondrick, and James Zou. 2025. Can LLM feedback enhance review quality? A randomized study of 20K reviews at ICLR 2025. *Arxiv preprint*.

Sowmya Vajjala and Shwetali Shimangaud. 2025. Text Classification in the LLM Era – Where do we stand? *Arxiv preprint*.

Hendrik P. van Dalen and Kène Henkens. 2012. Intended and unintended consequences of a publish-or-perish culture: A worldwide survey. *Journal of the American Society for Information Science and Technology*, 63(7):1282–1293.

Mark Ware and Michael Mabe. 2015. An overview of scientific and scholarly journal publishing. *The STM Report*, page 1082:1083.

Qiyao Wei, Samuel Holt, Jing Yang, Markus Wulfmeier, and Mihaela van der Schaar. 2025. The AI Imperative: Scaling High-Quality Peer Review in Machine Learning. *Arxiv preprint*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xiaotong (Tone) Xu, Jiayu Yin, Catherine Gu, Jenny Mar, Sydney Zhang, Jane L. E, and Steven P. Dow. 2024. Jamplate: Exploring LLM-Enhanced Templates for Idea Reflection. In *Proceedings of the 29th International Conference on Intelligent User Interfaces, IUI '24, March 18-21, 2024, Greenville, SC, USA*, page 907–921. Association for Computing Machinery.

Antoun Yaacoub, Zainab Assaghir, Lionel Prevost, and Jérôme Da-Rugna. 2025. Analyzing Feedback Mechanisms in AI-Generated MCQs: Insights into Readability, Lexical Properties, and Levels of Challenge. *Arxiv preprint*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2025. Qwen2.5 technical report. *Arxiv preprint*.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, et al. 2025. Yi: Open foundation models by 01.ai. *ArXiv preprint*.

A Appendix

A.1 Choice of LLMs and implementation details

Following Purkayastha et al. (2025), we select models that are privacy-preserving in accordance with ARR publication ethics guidelines (ACL Publication Ethics Committee, 2024) and of manageable sizes for practical deployment. All the LLMs are implemented using the vLLM library and huggingface transformers (Wolf et al., 2020).[20]. For calculating readability, we use the `textstat` package.[21] The experiments are run on a single A100 GPU and none of the experiments consumed more than 36 hours.

A.2 Annotation Guidelines

A.2.1 Task Description

As the volume of scientific publications grows, maintaining high-quality peer review becomes increasingly important. One common issue in reviews is the presence of non-specific reviews, which makes it difficult for authors and program chairs to address concerns effectively. Your task is to identify such non-specific issues in paper reviews and classify them into predefined categories.

Due to the high reviewing load, the reviewers are often prone to different kinds of bias, which inherently contribute to lower reviewing quality and hampers scientific progress in the field. The ACL peer reviewing guidelines characterize some of such reviewing bias in the table below, known as lazy thinking. The table mentions the heuristic and the reason for the statement being problematic. In this task, we aim to annotate these biased sentences in peer-reviews from scientific conferences.

A.2.2 Typical *Lazy Thinking* and *Specificity Issues*

Some specificity and lazy thinking issues are shown in Table 5. The full *specificity* issues table is in Table 8 and *lazy thinking* in Table 9 respectively.

A.2.3 Outlining differences within Summary of Weaknesses and Comments, Suggestions, and Typos

Summary of Weaknesses: This field is often naively interpreted as the reasons to reject. ARR review guidelines provide the following definition:

[20] https://docs.vllm.ai/en/latest/

[21] https://pypi.org/project/textstat/

	Issue	Meaning / Phrasing
Lazy Thinking	Results are not novel	If the paper claims e.g., a novel method, and you think you've seen this before, you need to provide a reference.
	The topic is too niche	A main track paper may well make a big contribution to a narrow subfield.
Specificity	The contribution is not novel	Highly similar work X and Y has been published 3+ months prior to the submission deadline.
	X is not clear	Y and Z are missing from the description of X.

Table 5: Examples of peer review issues categorized as *lazy thinking* and *specificity* as per ACL ARR guidelines (Rogers and Augenstein, 2024).

> What are the concerns that you have about the paper that would cause you to favor prioritizing other high-quality papers that are also under consideration for publication? These could include concerns about correctness of the results or argumentation, limited perceived impact of the methods or findings (note that impact can be significant both in broad or in narrow sub-fields), lack of clarity in exposition, or any other reason why interested readers of *ACL* papers may gain less from this paper than they would from other papers under consideration. Where possible, please number your concerns so authors may respond to them individually.

This field should contain reasons that make the paper not ready for publication at the current stage.

Comments, Suggestions, and Typos: These are additional suggestions that reviewers can provide to authors to improve the manuscript. ARR review guidelines define this field as:

> If you have any comments to the authors about how they may improve their paper, other than addressing the concerns above (weakness), please list them here.

The major difference between these two fields is the **severity of the issue**.

A.2.4 Annotation Instructions for Review Sentences

Method	Model	Prec.	Rec.	F1
Sequential	Yi	0.72	0.71	0.72
	Phi	0.73	0.72	0.72
	Qwen	0.67	0.61	0.64
	Deep.	0.62	0.62	0.62
	Oss.	0.71	0.68	0.69
Standalone	Yi	0.78	0.76	0.77
	Phi	**0.81**	**0.77**	**0.79**
	Qwen	0.73	0.67	0.70
	Deep.	0.72	0.66	0.69
	Oss.	0.75	0.73	0.73

Table 6: Performance comparison of various models across different methods on the segment detection task, evaluated in terms of Precision (Prec.), Recall (Rec.), and F1.

To identify review segments, we first determine the boundary of each sentence using BIO tags. Given a review sentence, we classify it as the Beginning (B), Inside (I) of a review segment, or Other (O). We consider the preceding sentence: if it relates to a different topic, the current sentence is marked as B; if it continues the previous sentence, it is marked as I; and if it is not relevant (e.g., punctuation or non-substantial content), it is marked as O.

Next, we identify the section of the review sentence. Each sentence is categorized as belonging to the Summary of Weaknesses, Comments and Suggestions, or Summary of Strengths. According to ARR guidelines, major criticisms that influence acceptance decisions are placed in the Summary of Weaknesses, while other suggestions or minor criticisms belong in Comments and Suggestions.

Finally, we identify lazy thinking or non-specific writing using multi-label classification. For each sentence, we assign appropriate categories of lazy thinking (e.g., "the results are not novel") and non-specificity (e.g., "X was done in the way Y"). The entire segment, formed by all sentences tagged with B/I together, is considered before assigning labels. For example, the sentences "I do not understand the setup" (B) and "There is no mention of epochs or hyperparameters in the dataset" (I) together form a segment:

> "I do not understand the setup. There is no mention of epochs or hyperparameters in the dataset."

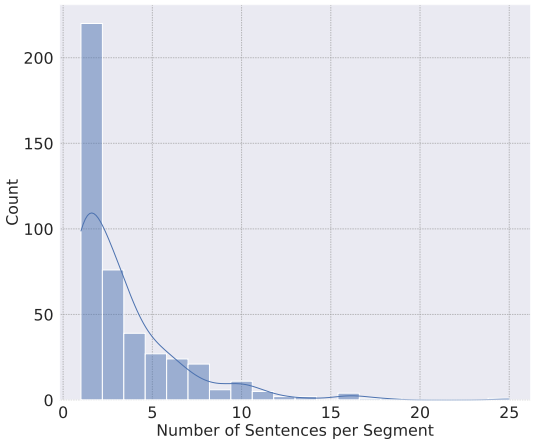
The segment collectively contributes to the label, e.g., **None**.

A.3 Analysis of the dataset, LAZYREVIEWPLUS

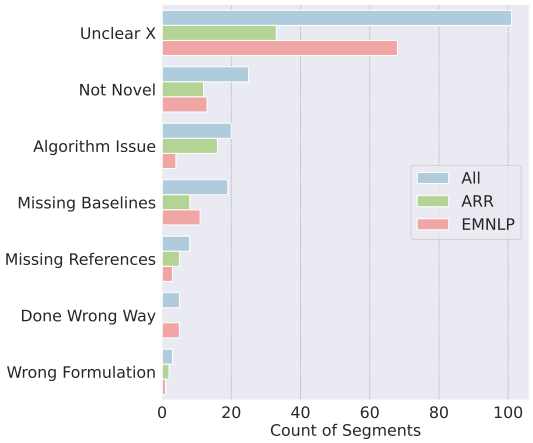
We show instances from our dataset in Table 7. We show the distribution of sentences per segment in Fig 2a.The distribution of labels within each segment in Fig 2c. The distribution of *specificity* and *lazy thinking* issues in Figures 2b and 2d respectively.

A.4 Segment Identification

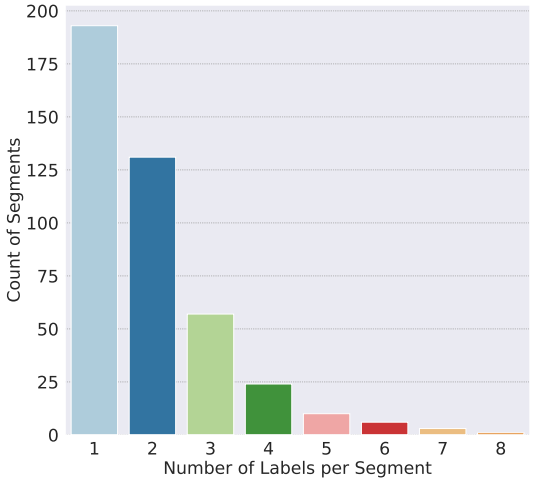
Task definition. For a review $R = (s_1, \ldots, s_n)$, each sentence s_i is tagged $y_i \in \{B, I, O\}$, where B marks a segment beginning, I an inside sentence, and O one outside any segment. The tagging function $f(s_i) = y_i$ yields $Y = (y_1, \ldots, y_n) = f(R)$.
Approach Let f_θ denote the LLM mapping a sentence (optionally with prior information) to a tag in B, I, O. Following prior work (Lan et al., 2024; Pichler et al., 2025), we perform segment detection using two approaches: **1. Standalone**: For each sentence s_i in review R, predict $\hat{y}i = f\theta(s_i, R)$ to indicate beginning (B), inside (I), or outside (O) a segment. This relies solely on s_i contextualized within the full review. **2. Sequential**: To capture inter-sentence dependencies, the LLM conditions on prior predictions $\hat{Y} < i = (\hat{y}1, \ldots, \hat{y}i - 1)$, predicting $\hat{y}i = f\theta(s_i, R, \hat{Y} < i)$. This provides temporary memory of past tagging decisions.
Overall Results. We present the performance of all models across different approaches in Table 6. The **standalone** strategy, which contextualizes each review segment within the full review, emerges as the most effective. In contrast, the **sequential** strategy suffers from error propagation, as reliance on prior predictions leads to cascading misclassifications, and sporadically may also introduces spurious correlations, consistent with prior findings (Feng et al., 2023; Purkayastha et al., 2025). Among the models, Phi consistently outperforms the others, which can be attributed to its specialized training on textbook-quality corpora combined with a balanced mix of synthetic and real-world data (Abdin et al., 2024).
Error Analysis. We analyze the performance of all models using the best-performing approach, standalone. We observe that the top model, Phi, occasionally confuses B tags with I tags. This pattern of confusion is consistent across the other models in our study (cf. Fig 4). This behavior is intuitive, as some review comments can be interpreted as



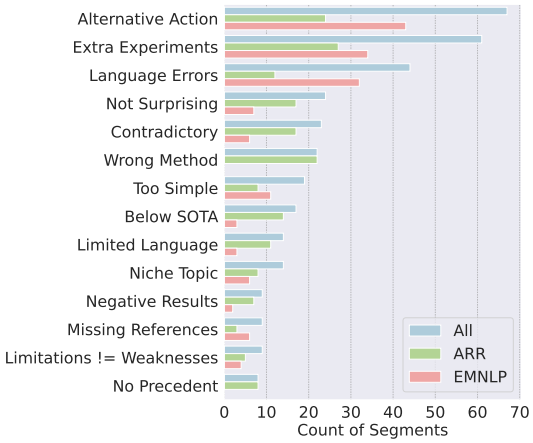
(a) Sentences within a segment



(b) Specificity Issues



(c) Distribution of labels within segments



(d) Distributions of labels within review segments and issues

Figure 2: Overview of sentence and label distributions in our proposed dataset, LAZYREVIEWPLUS. Issue names have been rewritten for brevity.

Review Segment	*Lazy Thinking*	*Specificity*
More experiments on datasets are needed, and I would also like to see some other LLMs, like the Gemini family, also be evaluated to make the paper more comprehensive and generalizable.	Authors could also do extra experiment; Authors should compare to a closed model	N/A
Although the paper is interesting and strongly supported by experimental results, understanding the writing is tough.	The paper has langauge errors	X is not clear
Results in Table 1 are lower than SOTA results, casting doubt on the usefulness and novelty of this approach.	Results do not surpass the latest SOTA; Results are not novel	The contribution is not novel

Table 7: Instances from our dataset, LAZYREVIEWPLUS where each segment may have multiple *lazy thinking* and *specificity* issues as per the ACL ARR guidelines (Rogers and Augenstein, 2024)

Non-Specific Issues	Specific Rewrite
The paper is missing relevant references	The paper is missing references XYZ
X is not clear	Y and Z are missing from the description of X.
The formulation of X is wrong	The formulation of X misses the factor Y
The contribution is not novel	Highly similar work X and Y has been published 3+ months prior to submission deadline
The paper is missing recent baselines	The proposed method should be compared against recent methods X, Y and Z (see H14 below for requesting comparisons to 'closed' systems)
X was done in the way Y	X was done in the way Y which has the disadvantage Z
The algorithm's interaction with dataset is problematic	It's possible that when using the decoding (line 723) on the dataset 3 (line 512), there might not be enough training data to rely on the n-best list.[If reasonably well-known entities are discussed]

Table 8: Examples of *specificity* issues and their corresponding specific rewrites.

standalone statements instead of occurring within the broader context. Nevertheless, the presence of extra 'B' tags does not substantially hinder **issue detection**, since our segment-level annotations ensure that each segment still corresponds to annotator detected issues within the dataset.

A.5 Additional Information of Issue Identification

A.5.1 Zero-shot Issue Identification

The zero-shot results from the best-performing LLM, GPT-OSS, fail to capture several classes that are sufficiently frequent in our dataset (cf. Fig. 6). While this type of classification achieves high precision for some classes, it introduces substantially more errors than a controlled approach.

A.5.2 Data Split for Issue Identification

We show the distributions of labels within train and test by employing various methods in Fig 3. Random review-level splits result in poor label balance, whereas our approach achieves nearly the same label balance as the `sklearn` train-test split. However, the `sklearn` split does not account for real-world scenarios, where full reviews are used as input for issue identification. In contrast, our distribution-aware split maintains label balance for effective cross-validation while also reflecting realistic use cases. We also show the distance across various split methods in Tab;e 11.

A.5.3 Prompts for Issue Identification

We present the **issue identification** prompt in Fig 5. The feature extraction prompt extracts abstract-level feature and converts it into the feature question for the LLM to answer using the Feature QA

Heuristic	Why this is problematic
H1. The results are not surprising	Many findings seem obvious in retrospect, but this does not mean that the community is already aware of them and can use them as building blocks for future work. Some findings may seem intuitive but haven't previously been tested empirically.
H2. The results contradict what I would expect	You may be a victim of confirmation bias, and be unwilling to accept data contradicting your prior beliefs.
H3. The results are not novel	If the paper claims e.g. a novel method, and you think you've seen this before - you need to provide a reference (note the policy on what counts as concurrent work). If you don't think that the paper is novel due to its contribution type (e.g. reproduction, reimplementation, analysis) — please note that they are in scope of the CFP and deserve a fair hearing.
H4. This has no precedent in the existing literature	Papers that are more novel tend to be harder to publish. Reviewers may be unnecessarily conservative.
H5. The results do not surpass the latest SOTA	SOTA results are neither necessary nor sufficient for a scientific contribution. An engineering paper could also offer improvements on other dimensions (efficiency, generalizability, interpretability, fairness, etc.) If the authors do not claim that their contribution achieves SOTA status, the lack thereof is not an issue.
H6. The results are negative	The bias towards publishing only positive results is a known problem in many fields, and contributes to hype and overclaiming. If something systematically does not work where it could be expected, the community does need to know about it.
H7. This method is too simple	The goal is to solve the problem, not to solve it in a complex way. Simpler solutions are preferable, as they are less brittle and easier to deploy in real-world settings.
H8. The paper doesn't use [my preferred methodology]	NLP is an interdisciplinary field, relying on many kinds of contributions: models, resource, survey, data/linguistic/social analysis, position, and theory.
H9. The topic is too niche	A main track paper may well make a big contribution to a narrow subfield.
H10. The approach is tested only on [not English]	The same is true of NLP research that tests only on English. Monolingual work on any language is important practically (methods/resources) and theoretically (deeper understanding of language).
H11. The paper has language errors	As long as the writing is clear enough, better scientific content should be more valuable than better journalistic skills.
H12. The paper is missing the [reference X]	Per ACL policy, missing references to prior highly relevant work is a problem if such work was published 3+ months before the submission deadline. Otherwise, missing references belong in the "suggestions" section.
H13. The authors could also do [extra experiment X]	It is always possible to come up with extra experiments and follow-up work. But a paper only needs to present sufficient evidence for the claim that the authors are making. Extra experiments belong in the "nice-to-have" category rather than "reasons to reject."
H14. The authors should compare to a 'closed' model X	Requesting comparisons to closed-source models is only reasonable if it directly bears on the claim the authors are making. Many important questions require more openness than closed models allow.
H15. The authors should have done [X] instead	"I would have written this paper differently." This criticism applies only if the authors' choices prevent them from answering their research question or their framing is misleading. Otherwise, it is just a suggestion, not a weakness.
H16. Limitations != weaknesses	No paper is perfect, and most CL venues now require a Limitations section. A good review should not list limitations as reasons to reject unless appropriately motivated.

Table 9: *Lazy Thinking* issues as per ARR Reviewer guidelines (Rogers and Augenstein, 2024).

prompt. The response from the Feature QA prompt is mapped to the discrete values: Yes → 1, No → -1, and Other → 0. We show a running example of a segment and abstract feature questions in Fig 7.

A.5.4 Performance of different methods

We present the performance of various classifiers on the issue detection task using representations from the `all-MiniLM-L6-v2` sentence transformer model in Table 14, and compare it to our approach under a 50% train-test split in Table 16. The performance of classifiers using a feature vector with five questions on Phi is shown in Table 15. Precision and recall of the different baselines discussed in §4 are reported in Table 18.

A.5.5 Performance with human-written questions

We also experimented with human-written questions for each review. Crafting 10 questions per issue type required approximately 45 minutes, so scaling this to 27 issue types took 27 × 45 minutes = 20.25 hours. The results with human-written questions are provided in Table 20. Precision with Extra Trees using Phi achieves 0.57 F1-score as compared to 0.51 with LLM-generated questions (cf. Table 1). However, since review guidelines frequently change, our LLM-based generation approach is more economical and provides a sustainable way to handle such dynamic conditions. Thus, the small gain in performance from human-written questions does not outweigh the practical benefits of using an LLM-driven question generation approach.

A.6 Additional details on feedback generation

A.6.1 Alignment Study with `Prometheus` and other evaluators

To assess the reliability of our reference-free automated metric, we conducted an alignment study comparing its scores with those provided by `Prometheus` and human annotators. The results, summarized in Table 21, show an average Spearman correlation of **0.85**, demonstrating the strong effectiveness of the metric. We further compare the performance of `Prometheus` v1, `Prometheus` v2 and GPT 4o in Table 22. `Prometheus` V2 outperforms all the other methods across the board in terms of Kendall τ.

A.6.2 Experimental Setup and Analysis for review rewrites and role extension

To identify occurrences of review extension and role separation in LLM-generated feedback, we use GPT-OSS 120B (Agarwal et al., 2025) as a judge, prompting it to detect such issues within zero-shot generations using the instructions in Table 23. The alignment between GPT-OSS judgments and human evaluations yields a Spearman correlation of 0.85. Table 23 presents the results, illustrating the prevalence of role separation' and review extension' in most of LLM-generated feedback. Example cases of role separation and review extension are shown in Fig. 10 and Fig. 9, respectively.

A.6.3 List of forbidden terms in the fitness function

To ensure that feedback remains professional, actionable, and relevant, we identify greetings and off-topic expressions that should be avoided in review comments. Such terms do not contribute to the evaluation of the work and may distract or confuse the author. In our zero-shot feedback generations, we found that approximately 82% of the divergent feedback contained one or more of these terms. Consequently, we compiled them into a list of forbidden terms to promote professionalism and maintain focus in reviewer feedback. `forbidden_terms = ["Hi", "Hello", "Hey", "Dear Author", "To whom it may concern", "Greetings", "Good morning", "Good afternoon", "Good evening", "Haha", "Hehe", "Lmao", "OMG", "Wow", "FYI", "Cheers", "Best regards", "Sincerely", "I think"]`

A.6.4 Prompts for feedback generation

Prompt for `Prometheus` The Conciseness prompt is in Fig 11, Relevance is in Fig 12, Specificity is in Fig 14 and Constructiveness is in Fig 13 respectively.

`1-pass` or zero-shot. The prompt is provided in Fig 15.

Plan Generation. The plan generation prompt is in Fig 17.

Template (`Temp`). The prompt is provided in Fig 16

Plan-then-generate (`Plan`) The prompt is provided in Fig 18.

Self Refine. (`Self Ref.`) The `1-pass` prompt is first run to create the initial feedback. Then the self-refine prompt in Fig 19 is run n_{gen} times to get the final feedback.

Genetic Algorithm. The genetic algorithm first uses Plan-then-generate prompt to generate candidates in Fig 18 multiple times which are scored via

the fitness function. The top $n_{parents}$ selected then go throug crossover to generate new offsprings as in Fig 20. The new candidate pool then goes through n_{gen} iterations to generate the final candidate.

A.6.5 Hyper-parameters used

Following prior work on using evolutionary algorithm for generating responses (Lee et al., 2025; Pham et al., 2025). we set the hyper-parameters as in Table 24.

A.6.6 Evaluation of the generated feedback

We present the full results of the automated evaluation using `Prometheus` in Table 13 and human evaluation (cf. §5) of the feedback in Table 25.

A.6.7 Ablation Study Description for the genetic algorithm

To better understand the contributions of each component in our genetic algorithm for feedback generation, we conduct an ablation study, systematically removing one component at a time:

1. Without Template Construction: The algorithm generates feedback without using the author-crafted issue-specific templates. This tests the importance of structured scaffolds in guiding precise and relevant feedback.

2. Without Plan Generation: The planner module is disabled, so the LLM does not generate a knowledge-driven plan to enrich the template. Feedback relies only on the template and review segment.

3. Without Population Initialization: Multiple candidate feedbacks are not generated initially. Only a single feedback is produced per review segment, removing diversity from the initial population.

4. Without Fitness Evaluation: Candidate feedback is not scored or ranked. All generated feedback is treated equally, removing the model's ability to optimize for conciseness, relevance, and constructiveness.

5. Without Parent Selection & Crossover: Evolutionary steps are skipped, so no combination of high-quality candidates occurs. This evaluates the contribution of crossover in improving feedback quality and diversity.

6. Without Final Candidate Selection: The highest-scoring feedback is not selected at the end of the process. A candidate is chosen arbitrarily, which assesses the effect of selecting the optimal feedback.

We present the results of the ablation study in Table 26. Each component contributes meaningfully to the quality of the generated feedback, highlighting the effectiveness of our approach.

A.6.8 Ablation on the Planner Component

We ablate the planner using multiple knowledge sources—abstract, reviewer-written strengths, and reviewer-written summaries—in Table 27. Our results show that incorporating all components is essential for generating the most specific feedback.

A.6.9 Ablation study on the rewards used in the fitness function for feedback generation

We present the results of the ablation study on different rewards in Table 28. We find that template adherence is the key factor driving the generation of highly effective feedback. Further analysis is provided in §5.

A.7 Example feedback

We show an example feedback generated by our approach in Fig 22.

A.8 Latency Analysis and Scaling

We show best and worst case estimate of deploying our feedback generation system to a real-world conference reviewing setup.

Assumptions:

- Average review: 5–10 segments
- LLM Q&A per batch of 16 segments: 1–2 s
- Segmentation: 0.1 s
- Issue Detection: 0.05 s
- Feedback generation: 1–2 s per review

Per-review Latency:

- LLM Q&A latency: 1–2 s
- Issue Detection latency: 0.05 s
- Feedback generation latency: 1–2 s
- Segmentation latency: 0.1 s

Total end-to-end latency per review: $0.1 + (1–2) + 0.05 + (1–2) \approx 2.15–4.15$ s per review

Scaling to 5,000 Reviews (Single GPU, Sequential):

- Best case: $5,000 \times 2.15\,\text{s} = 10,750\,\text{s} \approx 2.99\,\text{hours}$
- Worst case: $5,000 \times 4.15\,\text{s} = 20,750\,\text{s} \approx 5.77\,\text{hours}$

Multi-GPU Parallelization:

$$\text{Total wall-clock time} \approx \frac{\text{Sequential time}}{\text{Number of GPUs}}$$

By leveraging `vLLM` with a batch size of 16, LLM-based feature extraction is reduced to 1–2 seconds per review, and the genetic algorithm adds only 1–2 seconds, resulting in an end-to-end latency of $\sim$2–4 seconds per review. For a high-volume scenario of 5,000 reviews, sequential processing on a single GPU would require $\sim$3–6 hours, while parallelization across 4–8 GPUs reduces wall-clock time to $\sim$20–90 minutes, making the framework practical for deployment in time-sensitive contexts such as conference reviewing.

Issue / Heuristic	Template
H1. The results are not surprising	Your comment — "[insert reviewer comment here]" — suggests that the result is not surprising. While this may reflect your expertise, we encourage you to provide evidence or references (e.g., [references similar to title of the paper]) if this outcome is already known. What feels intuitive may not have been previously established, and empirical confirmation can still be a valuable contribution.
H2. The results contradict what I would expect	Your comment — "[insert reviewer comment here]" — indicates that the results contradict your expectations. Please consider that this may reflect confirmation bias, where prior beliefs influence interpretation. It's important to evaluate empirical findings objectively, even if they challenge existing assumptions. We encourage you to focus on the evidence presented and discuss how these findings advance understanding, rather than dismissing them based on expectation.
H3. The results are not novel	Your comment — "[insert reviewer comment here]" — raises concerns about novelty. If you believe the work duplicates prior work, please provide specific references ([references similar to the paper]) to support this claim (keeping in mind policies on concurrent work). If the paper's contribution is a reproduction, reimplementation, or analysis, please note that these are within the scope of the call and deserve fair consideration. Clear justification will help ensure the review is constructive and fair.
H4. This has no precedent in the existing literature	Your comment — "[insert reviewer comment here]" — indicates that the work has no clear precedent. While novelty is valuable, please be aware that highly novel contributions can be challenging to publish and may initially seem unfamiliar. Reviewers are encouraged to carefully consider the potential impact of innovative ideas rather than dismissing them due to lack of precedent. Providing constructive suggestions on how the authors can better frame or validate their novel contribution would be helpful.
H5. The results do not surpass the latest SOTA	Your comment — "[insert reviewer comment here]" — highlights that the results do not surpass the latest state-of-the-art (SOTA). Please keep in mind that achieving SOTA is neither necessary nor sufficient for a scientific contribution. Contributions improving other important aspects like efficiency, generalizability, interpretability, or fairness are also valuable. If the authors do not claim SOTA status, the lack of it should not be considered a weakness.
H6. The results are negative	Your comment — "[insert reviewer comment here]" — suggests the results are negative. Please consider that the bias toward publishing only positive results is a known issue across many fields. Negative or null results that show where methods systematically do not work are important for the community to understand limitations and avoid hype or overclaiming. Acknowledging the value of such findings can help make your review more balanced and constructive.
H7. This method is too simple	Your comment — "[insert reviewer comment here]" — suggests the method is too simple. However, the goal is to solve the problem effectively, not to add unnecessary complexity. Simpler methods often have advantages such as greater robustness, easier deployment, and better interpretability. Encouraging recognition of these benefits will help make your review more balanced and useful.
H8. The paper doesn't use [my preferred methodology]	Your comment — "[insert reviewer comment here]" — suggests a preference for a specific methodology (e.g., deep learning). Please note that NLP is an interdisciplinary field that values diverse contributions, including models, resources, surveys, data/linguistic/social analyses, theoretical work, and more. Limiting evaluation only to one methodology may overlook important advances. Recognizing this diversity will help make your review more comprehensive and fair.
H9. The topic is too niche	Your comment — "[insert reviewer comment here]" — suggests the topic is too niche. While the focus may be narrow, contributions to specific subfields can have significant impact and serve as important building blocks for the broader community. Recognizing the value of focused work helps ensure fair and balanced reviews.
H10. The approach is tested only on [not English]	Your comment — "[insert reviewer comment here]" — notes that the approach is tested only on [not English]. Please consider that monolingual research on any language is important both practically (developing methods and resources for that language) and theoretically (contributing to broader linguistic understanding). The NLP community values such focused contributions. Recognizing this helps ensure fair evaluation of research beyond English.
H11. The paper has language errors	Your comment — "[insert reviewer comment here]" — points out language errors. While clear writing is important, please prioritize the scientific content and contributions over minor language issues. Many papers can be improved with editorial help, but this should not overshadow the paper's value. Focusing on the core scientific merits will make your review more balanced and constructive.
H12. The paper is missing the [reference X]	Your comment — "[insert reviewer comment here]" — points out missing references. Please note ACL policy requires missing references to published highly relevant prior work that appeared at least three months before the submission deadline to be addressed seriously. Missing references to unpublished or very recent preprints should be treated as suggestions rather than critical flaws. Clarifying this distinction will help maintain fair and policy-compliant reviews.
H13. The authors could also do [extra experiment X]	Your comment — "[insert reviewer comment here]" — suggests additional experiments. While extra experiments can always be proposed, a paper only needs to provide sufficient evidence to support its claims. Additional experiments are generally considered "nice-to-have" and fit better in the suggestions section rather than as reasons for rejection. If you believe an extra experiment is essential for validity, please clearly justify why in your review.
H14. The authors should compare to a 'closed' model X	Your comment — "[insert reviewer comment here]" — requests comparisons to closed-source models (e.g., ChatGPT). Please consider that such comparisons are only meaningful if they directly support the paper's claims. Issues like test contamination and lack of transparency often limit their usefulness. Scientific progress often relies on openness and reproducibility, which closed models do not readily provide. Encouraging openness helps maintain rigorous evaluation standards.
H15. The authors should have done [X] instead	Your comment — "[insert reviewer comment here]" — suggests the authors should have done [X] instead. While multiple valid approaches often exist, this criticism is only warranted if the authors' choices prevent them from answering their research question, if their framing is misleading, or if the question itself lacks merit. Otherwise, such remarks are best framed as suggestions rather than weaknesses. Clarifying this distinction helps make your review fairer and more constructive.
H16. Limitations != weaknesses	Your comment — "[insert reviewer comment here]" — treats the paper's acknowledged limitations as weaknesses. While it's important to recognize limitations, they do not by themselves justify rejection. Most CL venues expect a Limitations section to promote transparency. If you believe a limitation seriously undermines the work, please clearly explain why. Distinguishing limitations from fatal flaws will improve the fairness and usefulness of your review.
The paper is missing relevant references	Your comment — "[insert reviewer comment here]" — needs to be rewritten by mentioning the exact references the authors are missing.
X is not clear	Your comment — "[insert reviewer comment here]" — needs to be rewritten by specifying the details that are exactly missing such as Y and Z are missing from the description of X.
The formulation of X is wrong	Your comment — "[insert reviewer comment here]" — needs to be rephrased as the formulation of X misses the component Y.
The contribution is not novel	Your comment — "[insert reviewer comment here]" — needs to be rephrased as highly similar work X and Y has been published 3+ months prior to submission deadline.
The paper is missing recent baselines	Your comment — "[insert reviewer comment here]" — needs to be rephrased as the proposed method should be compared against recent methods X, Y and Z.
X was done in the way Y	Your comment — "[insert reviewer comment here]" — needs to be rephrased as X was done in the way Y which has the disadvantage Z.

Table 10: Templates corresponding to heuristics and common reviewer issue types.

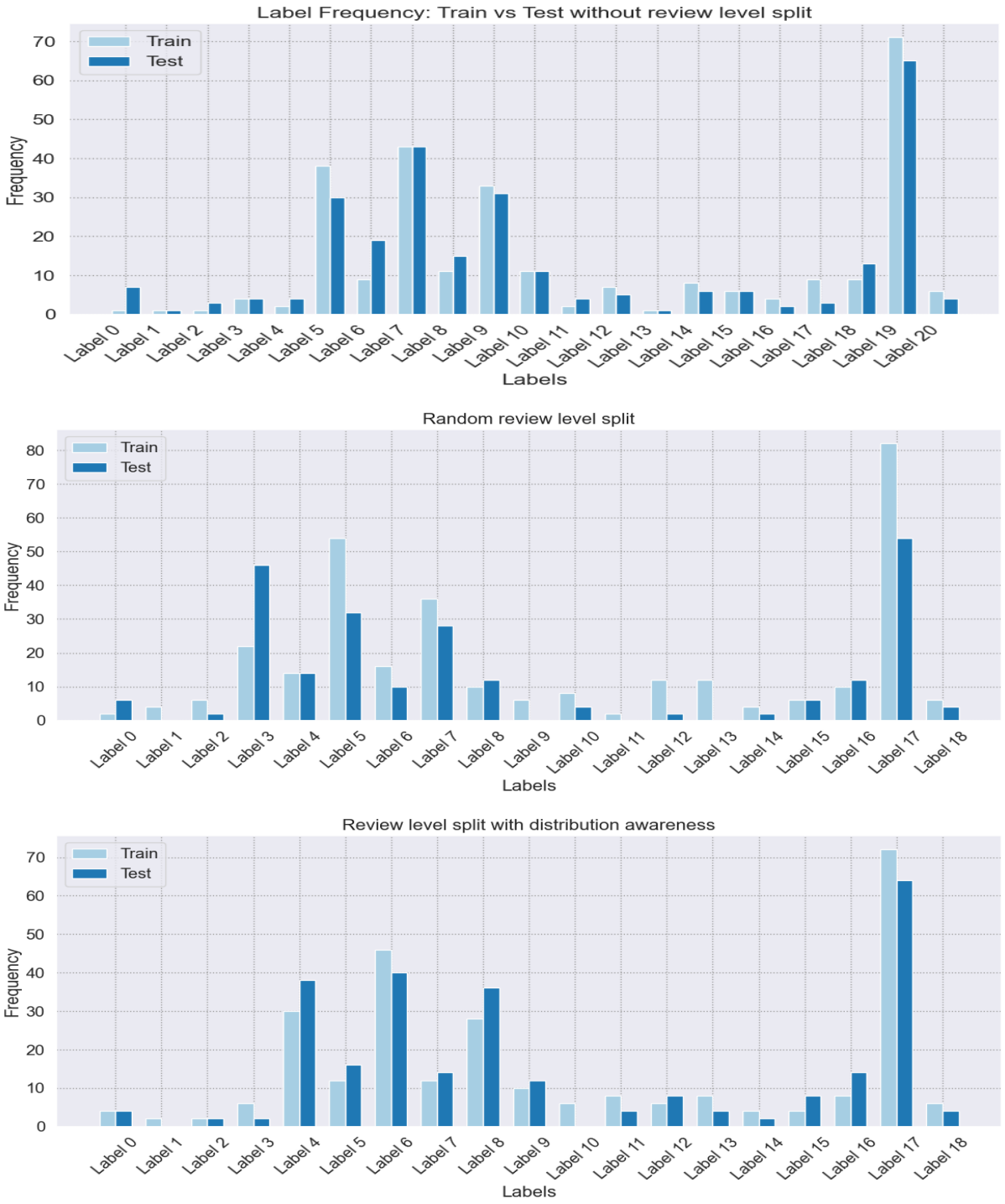


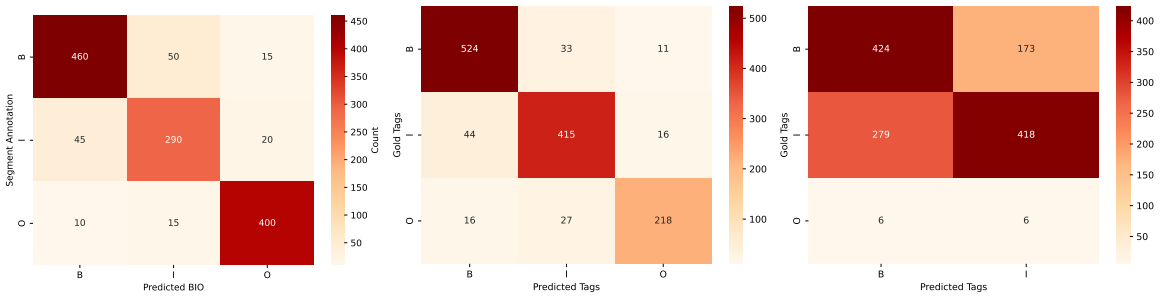
Figure 3: Comparing the label distribution using various split methods.

	Review-level-random-split	**Review-level-split**	**sklearn-split**
Distance	0.036	0.014	0.015

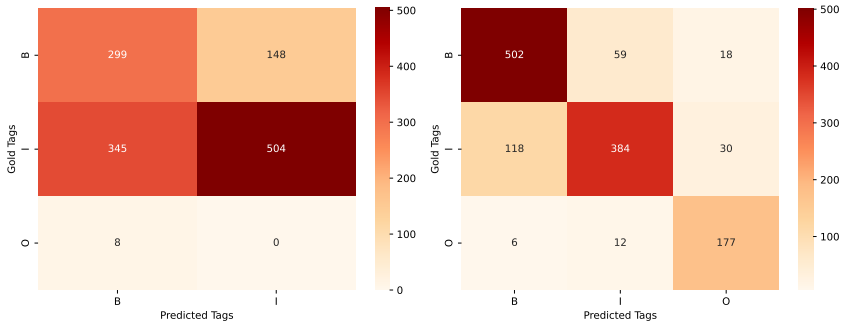
Table 11: Distance value for the three different split methods using evenly split 50% of data

Method	Yi		Phi		Qwen		Deep.		Oss.	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
KNN	0.51	0.27	0.43	0.23	0.44	0.21	0.41	0.18	0.44	0.20
Logistic Regression (L2)	0.52	0.32	0.52	0.30	0.46	0.33	0.39	0.29	0.41	0.28
Logistic Regression (L1)	0.49	0.35	0.46	0.34	0.40	0.34	0.42	0.31	0.42	0.32
Random Forest	0.78	0.19	0.79	0.21	0.74	0.16	0.66	0.15	0.65	0.18
Decision Tree	0.36	0.38	0.34	0.37	0.33	0.36	0.25	0.26	0.32	0.34
SVM (RBF)	0.83	0.15	0.79	0.19	0.69	0.12	0.77	0.13	0.66	0.12
SVM (Linear)	0.43	0.34	0.45	0.33	0.39	0.34	0.36	0.33	0.34	0.31
Gradient Boosting	0.52	0.31	0.54	0.37	0.49	0.30	0.41	0.23	0.46	0.29
AdaBoost	0.47	0.33	0.52	0.38	0.39	0.29	0.39	0.29	0.40	0.29
Extra Trees	0.76	0.23	0.75	0.23	0.70	0.18	0.68	0.17	0.64	0.19
MLP	0.51	0.31	0.56	0.33	0.44	0.33	0.49	0.28	0.47	0.27
Gaussian Naive Bayes	0.23	0.47	0.29	0.52	0.22	0.54	0.23	0.57	0.24	0.48

Table 12: Precision and Recall of different LLMs using our approach for **issue detection**.



(a) Yi (b) Phi (c) Qwen



(d) Deepseek (e) OSS

Figure 4: Confusion matrices for all the models on the **segment detection** task.

Issue Identification Prompt Library

Feature Extraction Prompt

You are given: - An issue label: issue - Why this issue is problematic: problem - Review segments (from NLP paper reviews): segments - Desired number of questions: 10
Your goal: Extract abstract, high level features that determine whether each review segment reflects the specified issue. Express each feature as an objective Yes/No question.
Rules you MUST follow: 1) Use ONLY the provided segments. Do NOT rely on external knowledge or assumptions. 2) Work at the ABSTRACT level (features that could generalize across wording), not surface phrasing. 3) Every question must be answerable by inspecting a single segment in isolation. 4) Questions must be neutral (no leading language), atomic (one idea per question), and non-redundant. 5) Avoid double negatives, multi-part questions, subjective terms ("clearly", "obviously"), or jargon unless it appears in the segments. 6) Prefer beginnings like "Does…", "Is…", "Are…", "Has…", "Do…", "Can…". 7) Keep each question concise (ideally 18 words). 8) Produce EXACTLY N questions. If N is not given, produce 10. 9) OUTPUT FORMAT: Return ONLY a Python list of strings, with double quotes, no code fences, no comments, no trailing commas. 10) Do NOT include any analysis, notes, or explanations in the output — only the list.
Step-by-step procedure: A) Parse and inventory: - Scan the segments to note common patterns about claims, evidence, specificity, comparisons, citations, quantification, assumptions, scope/coverage, and consistency. - Identify cues that would indicate presence/absence of the issue issue.
B) Abstract feature mining: - Convert recurring patterns into abstract properties that can reflect the similarity between the review segments you are given - Ensure each property directly supports diagnosing issue as problematic because problem.
C) Draft discriminative Yes/No tests: - For each property, write a Yes/No question that a reviewer could answer from a single segment. - Ensure questions collectively cover: evidence/citations, specificity/locating, correctness/faithfulness, scope, quantification, reproducibility/actionability, internal consistency, and fairness/balance (as applicable).
D) Prune and refine: - Remove overlaps; keep the most general, segment-checkable forms. - Rewrite to be atomic, neutral, and concise. - Ensure each question distinguishes "issue present" vs "issue absent".
E) Final checks (must pass all): - [ ] Exactly N questions. - [ ] All are Yes/No answerable from a single segment. - [ ] No duplicates or near-duplicates. - [ ] No restating issue verbatim; focus on testable properties. - [ ] Python list of strings ONLY, no extra text.
Now produce the final output.

Feature QA prompt

You will be given a review segment. Your task is to evaluate its quality by answering a series of Yes/No questions.
Review Segment: review
Question: question
Respond strictly with either:
[[Yes]] if the answer is Yes
[[No]] if the answer is No
[[Other]] if the question is irrelevant

Figure 5: Issue Identification Prompt

Method	Model	Const.	Relev.	Spec.	Conc.
1-pass	Yi	1.9	2.0	1.8	1.9
	Phi	2.1	2.2	2.0	2.1
	Qwen	1.8	1.9	1.7	1.8
	Deep.	1.7	1.8	1.6	1.7
	Oss.	1.9	2.0	1.8	1.9
BoN	Yi	2.0	2.2	2.4	2.3
	Phi	2.2	2.3	2.1	2.2
	Qwen	2.0	2.1	2.0	2.0
	Deep.	1.8	1.9	1.7	1.8
	Oss.	2.0	2.1	2.0	2.1
Self-Ref.	Yi	2.3	2.4	2.4	2.5
	Phi	2.4	2.6	2.8	2.9
	Qwen	2.2	2.4	2.4	2.7
	Deep.	2.4	2.4	2.5	2.3
	Oss.	3.0	3.1	2.9	3.0
Temp	Yi	2.8	2.9	2.7	2.8
	Phi	3.1	3.2	3.0	3.3
	Qwen	2.9	3.0	2.8	2.9
	Deep.	2.7	2.8	2.6	2.8
	Oss.	2.8	2.9	2.7	2.8
Plan	Yi	3.0	3.1	2.9	3.0
	Phi	3.3	3.4	3.2	3.5
	Qwen	3.1	3.2	3.0	3.1
	Deep.	2.9	3.0	2.8	3.0
	Oss.	3.0	3.1	2.9	3.0
Ours	Yi	3.9	3.8	3.8	3.8
	Phi	**4.3**	**4.3**	**4.2**	**4.3**
	Qwen	3.8	3.8	3.7	3.7
	Deep.	3.5	3.6	3.6	3.5
	Oss.	4.0	4.1	4.0	3.8

Table 13: Performance comparison of various models on the **feedback generation** task using four customized **automated metrics** from Prometheus V2: Constructiveness (Const.), Relevance (Relev.), Specificity (Spec.), and Conciseness (Conc.). Overall best results are **bolded** and method-wise best results are underlined.

Classifier	F0.5
KNN	0.28
Logistic Regression (L2)	0.39
Logistic Regression (L1)	0.39
Random Forest	0.10
Decision Tree	0.21
SVM (RBF)	0.22
SVM (Linear)	0.37
SVM (Poly)	0.00
AdaBoost	0.25
Extra Trees	0.09
Multi-layer Perceptron (MLP)	0.30
Gaussian Naive Bayes	0.34

Table 14: Performance of various classifiers on the **issue detection** task using representations obtained from all-MiniLM-L6-v2 sentence transformer model.

Classifier	F0.5	Precision	Recall
KNN	0.34	0.41	0.20
Logistic Regression (L2)	0.37	0.40	0.30
Logistic Regression (L1)	0.39	0.42	0.32
Random Forest	0.42	0.67	0.17
Decision Tree	0.34	0.33	0.38
SVM (RBF)	0.36	0.66	0.13
SVM (Linear)	0.34	0.34	0.34
SVM (Poly)	0.08	0.49	0.02
AdaBoost	0.43	0.47	0.32
Extra Trees	0.39	0.61	0.16
MLP	0.42	0.47	0.30
Gaussian Naive Bayes	0.22	0.19	0.57

Table 15: Performance of various classifiers using Phi-4 generated 5 question feature vector for the **issue detection** task.

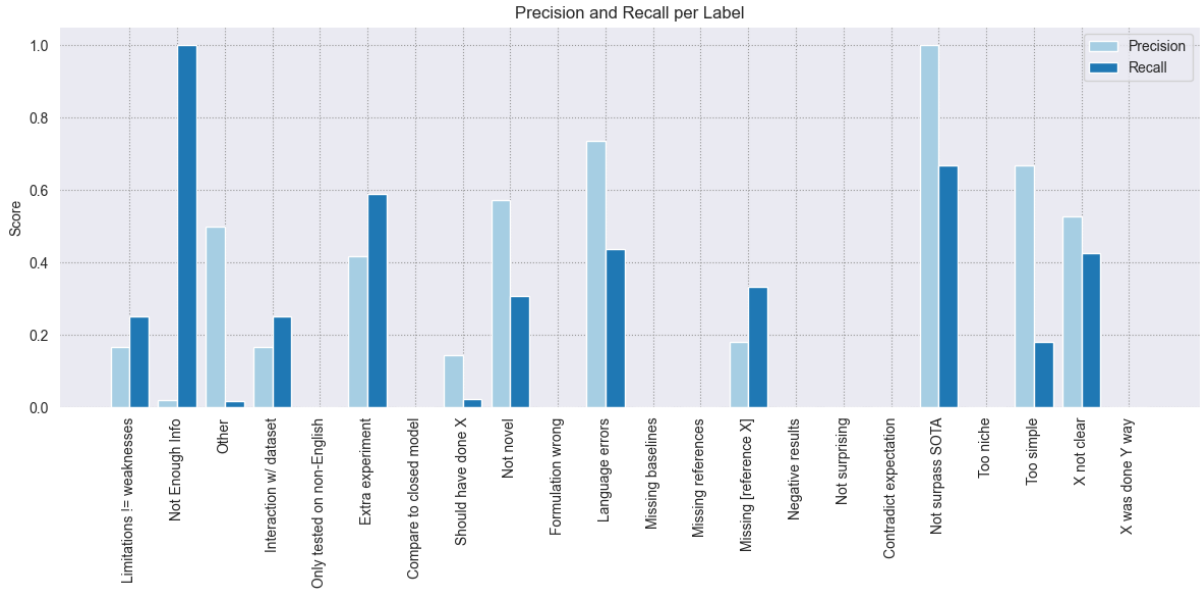


Figure 6: Zero shot results for **issue detection** with GPT oss 20B.

Label: The authors could also do [extra experiment X]

As described in the previous question, if it can be demonstrated that $\\lambda_2$ is significantly greater than $\\lambda_1$ when the data belongs to the C2 type, it would provide better interpretability for the results in Table 4.- The paper does not explore more complex causal graphs involving potential confounders and mediators.

The evaluation of the proposed method on test data consisting of both C1 and C2 reviews can be possible. First, an input review is classified into C1 or C2 by the peak-end rule. Then, the causal prompt C1 or C2, which corresponds to the classified type, is applied to predict a rating of the review. This system is compared with a baseline that always predicts a rating using a neutral prompt. Such an experiment can reveal the effectiveness of the proposed approach from more practical points of view.

...

→

Abstract Feature Questions:

Question 1: Does the reviewer suggest further tests, experiment or any kind of missing results without justify why they are necessary?

Question 2: ...

Figure 7: Example of abstract feature questions generated by our **issue detection** approach.

Methods	Ours. train	Ours. test	Emb. train	Emb. test
KNN	0.36	0.37	0.40	0.19
Logistic Regression (L2)	0.42	0.43	0.47	0.18
Logistic Regression (L1)	0.38	0.41	0.42	0.18
Random Forest	0.40	0.45	0.40	0.22
Decision Tree	0.34	0.37	0.28	0.19
SVM (RBF)	0.25	0.39	0.46	0.19
SVM (Linear)	0.36	0.40	0.45	0.17
SVM (Poly)	0.02	0.09	0.34	0.32
Gradient Boosting	0.36	0.41	0.33	0.21
AdaBoost	0.42	0.39	0.33	0.17
Extra Trees	0.37	0.44	0.46	0.28
Multi-layer Perceptron (MLP)	0.39	0.40	0.43	0.18
Gaussian Naive Bayes	0.30	0.31	0.42	0.18
Stochastic Gradient Descent (SGD)	0.35	0.43	0.38	0.20

Table 16: F0.5 score on even split of training and test with 50% each for **issue detection**. Embedding (Emb.) corresponds to representations obtained from `all-MiniLM-L6-v2` sentence transformer model.

Model	0.25			0.5			0.75			1.0			2.0		
	$F_{0.25}$	P	R	$F_{0.50}$	P	R	$F_{0.75}$	P	R	$F_{1.0}$	P	R	$F_{2.0}$	P	R
KNN	0.56	0.81	0.12	0.44	0.58	0.23	0.39	0.51	0.31	0.37	0.37	0.4	0.5	0.2	0.6
Logistic Regression (L2)	0.58	0.64	0.26	0.50	0.59	0.33	0.46	0.49	0.46	0.46	0.46	0.5	0.5	0.3	0.6
Logistic Regression (L1)	0.65	0.81	0.19	0.52	0.65	0.32	0.48	0.55	0.39	0.46	0.52	0.4	0.5	0.3	0.6
Random Forest	0.74	0.86	0.23	0.61	0.76	0.35	0.55	0.64	0.44	0.53	0.59	0.5	0.6	0.4	0.7
Decision Tree	0.35	0.35	0.38	0.35	0.35	0.38	0.36	0.35	0.38	0.36	0.35	0.4	0.4	0.3	0.4
SVM (RBF)	0.72	0.90	0.17	0.56	0.67	0.33	0.50	0.62	0.38	0.49	0.51	0.5	0.6	0.3	0.7
SVM (Linear)	0.54	0.74	0.15	0.45	0.51	0.33	0.43	0.44	0.42	0.43	0.39	0.5	0.5	0.3	0.7
Gradient Boosting	0.61	0.69	0.24	0.53	0.60	0.36	0.50	0.53	0.46	0.49	0.49	0.5	0.6	0.4	0.6
AdaBoost	0.59	0.66	0.28	0.51	0.59	0.33	0.47	0.51	0.44	0.48	0.44	0.6	0.5	0.3	0.7
Extra Trees	0.75	0.90	0.24	0.62	0.76	0.39	0.57	0.70	0.44	0.54	0.65	0.5	0.6	0.4	0.7
Multi-layer Perceptron (MLP)	0.67	0.78	0.22	0.56	0.66	0.36	0.50	0.64	0.38	0.47	0.57	0.4	0.5	0.3	0.6

Table 17: Performance of different models across thresholds 0.25, 0.5, 0.75, 1.0, and 2.0. Metrics shown are $F_{\text{threshold}}$ (e.g., $F_{0.25}$ for threshold 0.25), P (Precision), and R (Recall).

Baselines	Yi		Phi		Qwen		Deep.		Oss.	
	P	R	P	R	P	R	P	R	P	R
$\texttt{LLM}_{\texttt{zero}}$	0.03	0.12	0.09	0.11	0.06	0.10	0.03	0.10	0.22	0.24
$\texttt{LLM}_{\texttt{Fine}}$	0.23	0.23	0.11	0.18	0.13	0.17	0.10	0.19	0.19	0.21
$\texttt{LLM}_{\texttt{QA}}$	0.10	0.06	0.10	0.28	0.16	0.22	0.19	0.26	0.10	0.40
$\texttt{LLM}_{\texttt{QAFine}}$	0.10	0.20	0.09	0.27	0.05	0.12	0.07	0.19	0.10	0.17

Table 18: Precision (P) and Recall (R) of various LLMs for the **issue detection** task.

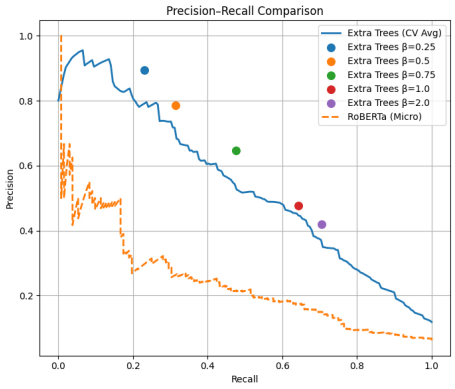


Figure 8: Precision-Recall Curve for RoBERTa and our approach operating at different thresholds.

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Variance
$F_{0.5}$	0.55	0.55	0.42	0.52	0.56	0.002
Precision	0.74	0.88	0.64	0.69	0.81	0.007
Recall	0.27	0.22	0.18	0.26	0.25	0.0009

Table 19: Cross-validation results over 5 folds showing $F_{0.5}$, Precision, Recall, and their variances.

Model	Mean $F_{0.5}$	Mean Precision	Mean Recall
KNN	0.544	0.667	0.314
Logistic Regression (L2)	0.514	0.576	0.365
Logistic Regression (L1)	0.456	0.477	0.391
Random Forest	0.554	0.761	0.270
SVM (RBF)	0.463	0.948	0.153
Extra Trees	0.575	0.783	0.281

Table 20: Mean $F_{0.5}$, precision, and recall across different models using human-written questions as features.

Review Extension

Original Comment: "...no check for false negatives... only one language, one domain, one genre... no significance testing." **Issue Detected**: **The authors could also do extra experiment, X**
Generated feedback:
You can rewrite your comment as the follows: While the paper presents a promising benchmark and methodology, the evaluation appears limited in scope. The experiments focus solely on English data from a single domain and genre, which may restrict the generalizability of the results. Including experiments across multiple languages or domains could strengthen the empirical claims. Additionally, incorporating significance testing would help confirm that the observed improvements are statistically meaningful. Overall, expanding the evaluation would improve both the rigor and the impact of the work.

Figure 9: Review Extension: Generating full reviews instead of feedback

Model	Constructiveness		Relevance		Specificity		Conciseness	
	Kendall τ	Spearman ρ	Kendall τ	Spearman ρ	Kendall τ	Spearman ρ	Kendall τ	Spearman ρ
Yi	0.75	0.83	0.76	0.84	0.75	0.83	0.76	0.85
Phi	0.78	0.88	0.77	0.87	0.76	0.86	0.78	0.88
Qwen	0.76	0.84	0.75	0.85	0.76	0.84	0.75	0.84
Deep.	0.75	0.83	0.75	0.84	0.75	0.83	0.75	0.83
Oss.	0.77	0.85	0.76	0.85	0.76	0.85	0.77	0.86
Average	0.76	0.85	0.76	0.85	0.76	0.84	0.76	0.85

Table 21: Kendall τ and Spearman ρ correlations with Prometheus for different models across four metrics: Constructiveness, Relevance, Specificity, and Conciseness. Method-wise best correlations are underlined. The last row shows the average correlation across all models and metrics.

	Constr.			Rel.			Spec.			Conc.		
Model	Pr-v1	GPT-4o	Pr-v2	Pr-v1	GPT-4o	Pr-v2	Pr-v1	GPT-4o	Pr-v2	Pr-v1	GPT-4o	Pr-v2
Yi	0.73	0.76	**0.75**	0.74	0.75	**0.76**	0.73	0.74	**0.75**	0.74	0.75	**0.76**
Phi	0.77	0.78	**0.78**	0.76	0.77	**0.77**	0.75	0.76	**0.76**	0.77	0.77	**0.78**
Qwen	0.75	0.75	**0.76**	0.74	0.75	**0.75**	0.75	0.75	**0.76**	0.74	0.75	**0.75**
Deep.	0.74	0.74	**0.75**	0.74	0.74	**0.75**	0.74	0.74	**0.75**	0.74	0.74	**0.75**
Oss.	0.76	0.76	**0.77**	0.75	0.75	**0.76**	0.75	0.75	**0.76**	0.76	0.76	**0.77**
Avg	0.75	0.76	**0.76**	0.75	0.76	**0.76**	0.74	0.75	**0.76**	0.75	0.76	**0.76**

Table 22: Kendall τ correlations across review quality dimensions for Prometheus v1 (Pr-v1), v2 (Pr-v2) and GPT-4o respectively.

Model	Review Ext. (%)	Role Sep. (%)
Yi	42	23
DeepSeek	47	30
OSS	43	28
Qwen	45	25
Phi	**41**	**23**

Table 23: Comparison of review generation issues across models. Review Extension refers to instances where the LLM generates full reviews instead of targeted feedback, while Role Separation indicates cases where feedback is misdirected to authors instead of reviewers. The least of the issues is highlighted

Hyper-parameter	Description	Value / Setting
T	Number of author-crafted templates	25
n	Number of initial candidate feedbacks per segment	10
n_{parents}	Number of parents selected per generation for crossover	5
τ	Temperature parameter for Boltzmann selection	0.1
$n_{\text{generations}}$	Number of evolutionary generations	3
n_{sent}	Number of maximum sentences allowed	5

Table 24: Hyper-parameters used in the genetic algorithm for **feedback generation** algorithm.

Method	Const.	Relev.	Spec.	Conc.
1-pass	1.8	1.9	2.2	2.2
BoN	2.4	2.4	2.5	2.5
Self-Ref.	2.5	2.6	2.6	2.6
Temp	2.8	2.7	2.8	2.8
Plan	3.2	3.1	3.2	3.2
Ours	**4.1**	**4.2**	**4.0**	**4.3**

Table 25: Human Evaluation of the best performing model Phi on the **feedback generation** task using Constructiveness (Const.), Relevance (Relev.), Specificity (Spec.) and Conciseness (Conc.)

Role Separation (Author-Facing Feedback)

Original Comment: "...no check for false negatives… only one language, one domain, one genre… no significance testing."
Issue Detected: **The authors could also do extra experiment, X**
Generated feedback:
The current experiments demonstrate the benchmark and methodology effectively, but they are limited to English, a single domain, and one genre. To strengthen the claims, consider including additional languages, domains, or genres where feasible. Also, performing significance testing would provide stronger evidence for the reported improvements. While these are suggestions to enhance the work, the main contribution remains valid and valuable within the presented scope.

Figure 10: Role Separation: Generating feedback to authors instead of reviewers

Model	Variant	Const.	Relev.	Spec.	Conc.
Yi	Full Algorithm (Ours)	**3.9**	**3.8**	**3.8**	**3.8**
	w/o Template Construction	3.6	3.5	3.4	3.5
	w/o Plan Generation	3.8	3.8	3.8	3.8
	w/o Population Initialization	3.9	3.9	3.7	3.7
	w/o Fitness Evaluation	3.7	3.7	3.7	3.7
	w/o Parent Selection & Crossover	3.9	3.8	3.7	3.7
	w/o Final Candidate Selection	3.8	3.7	3.6	3.7
Qwen	Full Algorithm (Ours)	**3.8**	**3.8**	**3.7**	**3.7**
	w/o Template Construction	3.5	3.5	3.4	3.4
	w/o Plan Generation	3.7	3.7	3.6	3.6
	w/o Population Initialization	3.8	3.8	3.6	3.6
	w/o Fitness Evaluation	3.6	3.6	3.6	3.5
	w/o Parent Selection & Crossover	3.7	3.7	3.6	3.6
	w/o Final Candidate Selection	3.6	3.6	3.5	3.5
DeepSeek	Full Algorithm (Ours)	**3.5**	**3.6**	**3.6**	**3.5**
	w/o Template Construction	3.2	3.3	3.2	3.2
	w/o Plan Generation	3.4	3.5	3.5	3.4
	w/o Population Initialization	3.5	3.5	3.4	3.4
	w/o Fitness Evaluation	3.3	3.3	3.3	3.3
	w/o Parent Selection & Crossover	3.4	3.4	3.3	3.3
	w/o Final Candidate Selection	3.3	3.3	3.2	3.3
Oss.	Full Algorithm (Ours)	**4.0**	**4.1**	**4.0**	**3.8**
	w/o Template Construction	3.7	3.8	3.7	3.6
	w/o Plan Generation	3.9	4.0	3.9	3.7
	w/o Population Initialization	4.0	4.0	3.9	3.8
	w/o Fitness Evaluation	3.8	3.9	3.8	3.7
	w/o Parent Selection & Crossover	3.9	3.9	3.8	3.7
	w/o Final Candidate Selection	3.8	3.8	3.7	3.7
Phi	Full Algorithm (Ours)	**4.3**	**4.3**	**4.2**	**4.3**
	w/o Template Construction	3.6	3.5	3.4	3.5
	w/o Plan Generation	3.8	3.8	3.8	3.8
	w/o Population Initialization	3.9	3.9	3.7	3.7
	w/o Fitness Evaluation	3.7	3.7	3.7	3.7
	w/o Parent Selection & Crossover	3.9	3.8	3.7	3.7
	w/o Final Candidate Selection	3.8	3.7	3.6	3.7

Table 26: Ablation study of various components within the genetic algorithm across the four automated metrics on **feedback generation** for all models.

Model	Variant	Const.	Relev.	Spec.	Conc.
Yi	Ours (Abstract + Summary + Strengths)	**3.9**	**3.8**	**3.8**	**3.8**
	w/o Abstract	3.6	3.7	3.5	3.6
	w/o Summary	3.7	3.7	3.6	3.7
	w/o Strengths	3.8	3.8	3.6	3.7
	w/o Abstract + Summary	3.5	3.6	3.4	3.5
	w/o Abstract + Strengths	3.6	3.7	3.5	3.6
Qwen	Ours (Abstract + Summary + Strengths)	**3.8**	**3.8**	**3.7**	**3.7**
	w/o Abstract	3.5	3.5	3.4	3.4
	w/o Summary	3.6	3.6	3.5	3.5
	w/o Strengths	3.7	3.7	3.5	3.6
	w/o Abstract + Summary	3.4	3.5	3.3	3.4
	w/o Abstract + Strengths	3.5	3.6	3.4	3.5
DeepSeek	Ours (Abstract + Summary + Strengths)	**3.5**	**3.6**	**3.6**	**3.5**
	w/o Abstract	3.2	3.3	3.2	3.2
	w/o Summary	3.3	3.4	3.3	3.3
	w/o Strengths	3.4	3.5	3.3	3.4
	w/o Abstract + Summary	3.1	3.2	3.1	3.2
	w/o Abstract + Strengths	3.2	3.3	3.2	3.2
Oss.	Ours (Abstract + Summary + Strengths)	**4.0**	**4.1**	**4.0**	**3.8**
	w/o Abstract	3.7	3.8	3.7	3.6
	w/o Summary	3.8	3.9	3.8	3.7
	w/o Strengths	3.9	4.0	3.8	3.7
	w/o Abstract + Summary	3.6	3.8	3.5	3.6
	w/o Abstract + Strengths	3.7	3.9	3.6	3.7
Phi	Ours (Abstract + Summary + Strengths)	**4.1**	**4.2**	**4.0**	**4.3**
	w/o Abstract	3.8	4.0	3.7	4.1
	w/o Summary	3.9	4.0	3.8	4.0
	w/o Strengths	4.0	4.1	3.8	4.1
	w/o Abstract + Summary	3.6	3.9	3.5	3.9
	w/o Abstract + Strengths	3.7	3.9	3.6	4.0
	w/o Summary + Strengths	3.7	3.9	3.6	4.0

Table 27: Planner ablation study across the four automated metrics on **feedback generation** for all models.

Conciseness Evaluation Prompt

Task Description: You are given an instruction, a response to evaluate, and a score rubric representing an evaluation criterion.
Your goal: 1. Write detailed feedback that assesses the response strictly based on the rubric. 2. Assign a score between 1 and 5, referring to the rubric. 3. Output in the following format:

> Feedback: (write a feedback for criteria) [RESULT] (an integer number between 1 and 5)

4. Do not include any opening, closing, or explanations.
Instruction to evaluate: You are an expert evaluating feedback generated for improving review segments. The feedback provided will help the reviewer improve the segment.
Weakness: weakness
Feedback to evaluate: feedback
Score Rubric: [Does the feedback communicate suggestions concisely? Feedback should be brief, precise, and to the point, avoiding unnecessary verbosity while remaining actionable.]

- Score 1: Feedback is wordy, repetitive, or confusing; hard to extract actionable guidance.
- Score 2: Feedback conveys ideas but is often verbose, vague, or partially actionable; contains unnecessary wording.
- Score 3: Feedback is moderately concise, understandable, and partially actionable; some redundancy or extra wording may remain.
- Score 4: Feedback is clear, focused, and mostly concise; communicates actionable guidance efficiently, with minor verbosity.
- Score 5: Feedback is precise, concise, and easy to interpret; provides direct, actionable suggestions covering both minor and substantive issues effectively.

Feedback:

Figure 11: `Prometheus` Conciseness Prompt

Relevance Evaluation Prompt

Task Description: You are given an instruction, a response to evaluate, and a score rubric representing an evaluation criterion.
Your goal: 1. Write detailed feedback that assesses the response strictly based on the rubric. 2. Assign a score between 1 and 5, referring to the rubric. 3. Output in the following format:

> Feedback: (write a feedback for criteria) [RESULT] (an integer number between 1 and 5)

4. Do not include any opening, closing, or explanations.
Instruction to evaluate: You are an expert evaluating feedback generated for improving review segments. The feedback provided will help the reviewer improve the segment.
Weakness: weakness
Feedback to evaluate: feedback
Score Rubric: [Is the feedback directly related to the content or structure of the review? Feedback should focus on aspects that help improve the review, including conciseness, reasoning, accuracy, or actionable corrections.]

- Score 1: Feedback is off-topic or unrelated to the review's content or structure; provides no actionable guidance.
- Score 2: Feedback mentions the review but mostly addresses minor or tangential points; limited actionable guidance.
- Score 3: Feedback is partially relevant; identifies some issues but mixes relevant and loosely connected commentary; limited guidance.
- Score 4: Feedback is clearly related to review content or structure; provides actionable suggestions addressing key issues.
- Score 5: Feedback is tightly focused on review content or structure; comments are precise, actionable, and justified, improving clarity, reasoning, or accuracy.

Feedback:

Figure 12: `Prometheus` Relevance Prompt

Constructiveness Evaluation Prompt

Task Description: You are given an instruction, a response to evaluate, and a score rubric representing an evaluation criterion.
Your goal: 1. Write detailed feedback that assesses the response strictly based on the rubric. 2. Assign a score between 1 and 5, referring to the rubric. 3. Output in the following format:

> Feedback: (write a feedback for criteria) [RESULT] (an integer number between 1 and 5)

4. Do not include any opening, closing, or explanations.
Instruction to evaluate: You are an expert evaluating feedback generated for improving review segments. The feedback provided will help the reviewer improve the segment.
Weakness: weakness
Feedback to evaluate: feedback
Score Rubric: [Does the feedback offer suggestions for how the reviewer can improve? Constructive feedback should guide revisions, not just point out flaws, and can address both minor textual issues and substantive content.]

- Score 1: Feedback only identifies flaws without suggesting improvements; unhelpful or dismissive.
- Score 2: Feedback includes vague or superficial advice; little actionable guidance.
- Score 3: Feedback identifies issues and offers some guidance but partially unclear or limited.
- Score 4: Feedback provides clear and relevant suggestions; mostly actionable, may lack some detail.
- Score 5: Feedback consistently identifies issues and provides specific, practical, and helpful suggestions; supports targeted improvements effectively.

Feedback:

Figure 13: `Prometheus` Constructiveness Prompt

Specificity Evaluation Prompt

Task Description: You are given an instruction, a response to evaluate, and a score rubric representing an evaluation criterion.
Your goal: 1. Write detailed feedback that assesses the response strictly based on the rubric. 2. Assign a score between 1 and 5, referring to the rubric. 3. Output in the following format:

> Feedback: (write a feedback for criteria) [RESULT] (an integer number between 1 and 5)

4. Do not include any opening, closing, or explanations.
Instruction to evaluate: You are an expert evaluating feedback generated for improving review segments. The feedback provided will help the reviewer improve the segment.
Weakness: weakness
Feedback to evaluate: feedback
Score Rubric: [Does the feedback refer to specific parts or issues in the review? Feedback should clearly indicate what needs improvement, avoiding vague statements without context.]

- Score 1: Feedback is entirely vague or generic; no reference to particular parts of the review; not actionable.
- Score 2: Feedback hints at an issue but lacks concrete references; reviewer cannot easily locate the problem.
- Score 3: Feedback identifies general areas or sections but does not pinpoint exact sentences or claims; partially actionable.
- Score 4: Feedback refers to specific sections or issues; minor vagueness may remain; mostly actionable.
- Score 5: Feedback clearly identifies exact parts of the review and provides precise, actionable guidance, including minor textual or deeper content issues when justified.

Feedback:

Figure 14: `Prometheus` Specificity Prompt

Lazy Thinking Issues

H1. The results are not surprising: Many findings seem obvious in retrospect, but this does not mean the community is already aware of them. Some findings may seem intuitive but haven't been empirically tested.
H2. The results contradict expectations: Reviewer may be biased against findings that challenge prior beliefs.
H3. The results are not novel: If a claimed novel method resembles existing work, provide references. Reproduction, reimplementation, or analysis papers may still be in scope.
H4. No precedent in existing literature: Novel papers are harder to publish; reviewers may be overly conservative.
H5. Results do not surpass latest SOTA: SOTA results are not required; other contributions (efficiency, fairness, interpretability) matter.
H6. Negative results: Publishing negative results is important; systematic failures provide valuable knowledge.
H7. Method too simple: Simple solutions are often preferable—less brittle and easier to deploy.
H8. Missing preferred methodology: NLP is interdisciplinary; multiple approaches (models, resources, analyses) are valid.
H9. Topic too niche: Significant contributions can be made to narrow subfields.
H10. Tested only on non-English: Monolingual studies are important for both practical and theoretical contributions.
H11. Language errors: Focus on scientific content; minor writing issues should not dominate.
H12. Missing reference(s): Prior work published >3 months before submission should be cited; preprints are optional.
H13. Extra experiments suggested: Optional experiments are "nice-to-have" and belong in suggestions, not as reasons to reject.
H14. Compare to closed model: Only meaningful if it directly affects the paper's claim; closed models often introduce methodological issues.
H15. Authors should have done X instead: Only relevant if choices prevent addressing the research question; otherwise a suggestion, not a weakness.
H16. Limitations != weaknesses: Limitations sections are standard; do not equate limitations with reasons to reject unless strongly justified.

Specificity Issues

Specificity Issues:
Missing relevant references: The review fails to mention key prior work (e.g., XYZ).
X is unclear: Important details (e.g., Y and Z) are missing from description.
Formulation of X is wrong: The definition or equation of X omits critical factors (e.g., Y).
Contribution not novel: Similar work (e.g., X, Y) has been published >3 months prior.
Missing recent baselines: Proposed methods should be compared to recent approaches X, Y, Z.
X done in way Y: Implementation choices lead to known limitations (e.g., disadvantage Z).

Task

Task: You are an expert at providing actionable feedback to improve a review segment. Given a review, the segment, and lazy thinking annotation, provide feedback that the reviewer can use to revise the segment.

Review Segment: review segment
Issue: issue
Feedback :

Figure 15: Zero shot Feedback generation prompt

Templatic Feedback Generation Prompt

Task: You are an expert at providing actionable feedback to improve the writing of review segments. Given a review segment and its identified issue, your goal is to produce feedback that the reviewer can use to revise the segment. You are also provided a **feedback template** for the identified issue.
Instructions: Adapt the provided feedback template to match the comment in the review segment. Ensure the feedback is actionable, specific, and targeted to improving the segment.
Review Segment: weakness
Identified Issue: identified issue
Feedback Template: template

Output: The feedback generated for this review segment, following the guidance of the template.

Figure 16: Prompt for Templatic Feedback Generation using Identified Issues.

Planner Prompt for Feedback Generation

System Instruction: You are a planning agent for feedback generation.
Input:

- **Review Comment:** {review}
- **Issue:** {issue}
- **Abstract of the paper:** {abstract}
- **Summary written by reviewer:** {summary}
- **Strengths written by reviewer:** {strengths}

Task:

1. For each label, identify which external knowledge (Abstract, Summary, Template) is needed based on the mapping.
2. Suggest how to use the available knowledge (e.g., quote from abstract, summarize key claim).
3. Provide an explanation for your choice of plan.

Output: A JSON list with the following structure:

- `plan`: brief instructions on how to use the knowledge
- `explanation`: reasoning behind the chosen plan

Example Output:

```
[
  {
    "plan": "Use summary to show that results differ from intuition",
   "explanation": "The summary provides key claims that can highlight why the results are surprising."
  },
  {
    "plan": "Use abstract + summary to check novelty",
   "explanation": "Both abstract and summary help determine whether the contribution is novel."
  }
]
```

Figure 17: Full Planner Prompt for Feedback Generation.

Feedback Generation Prompt

System: You are an expert at generating actionable feedback to improve review segments.
Instructions: Given: 1. A review segment (weakness) 2. Identified issue(s) 3. A feedback template for the issue(s) 4. A plan describing what to use as knowledge sources (abstract, summary, template) to guide feedback. 5. An explanation decsribing how to use the knowledge sources.
Your task: - Generate actionable feedback for the reviewer to improve the review segment. - Incorporate the plan into the template: adapt the template to match the specific review segment and issue. - Keep the feedback precise, relevant, and constructive. - Output only the feedback text, do not include any explanations or extra text.
Review Segment: weakness
- Identified Issue: identified issue
Feedback Template: template
Plan: plan **Explanation**: explanation
Example Output: "Feedback: The reviewer should clarify the novelty of the contribution by referencing prior work; following the template, highlight which claims are incremental and which are novel. [5]"

Figure 18: Feedback generation using Plan and Template

Self-Refine Feedback Generation Prompt

System: You are an expert at generating and refining actionable feedback for improving review segments.
Instructions: Given: 1. A review segment (weakness) 2. Identified issue(s) 3. An initial draft feedback
Your task: - Generate detailed feedback that helps the reviewer improve the review segment. - Critically evaluate the initial feedback and refine it to be more precise, constructive, and actionable. - Ensure the feedback is relevant to the identified issue(s) and avoids vague statements. - Output only the refined feedback text, do not include explanations, reasoning, or any extra text.
Variables: - Review Segment: weakness - Identified Issue: identified issue - Initial Feedback: initial feedback
Example Output: "Feedback: The reviewer should provide concrete examples illustrating missing baselines; clearly indicate which comparisons are necessary and why. [5]"

Figure 19: Self-refine feedback generation for review segments

Cross-Over Feedback Prompt (5 Parents)

System: You are an expert at generating high-quality feedback for review segments by synthesizing multiple sources.
Instructions: Given: 1. A review segment (weakness) 2. Identified issue(s) 3. Five parent feedback drafts
Your task: - Combine the key insights from all 5 parent feedbacks into a single, coherent feedback. - Ensure that the final feedback is precise, constructive, and actionable. - Retain the strongest points from each parent, remove redundancy, and resolve conflicting suggestions. - Feedback must directly address the identified issue(s) and help the reviewer improve the segment. - Output only the final synthesized feedback, no explanations or commentary.
Variables: - Review Segment: weakness - Identified Issue: identified issue - Parent Feedbacks: 1) parent1 2) parent2 3) parent3 4) parent4 5) parent5
Example Output: "Feedback: The reviewer should clarify the novelty of the contribution and reference prior work where necessary; highlight incremental contributions while maintaining clarity and specificity. [5]"

Figure 20: Cross-over feedback synthesis for 5 parent feedbacks

Review Extension and Role Separation Detection Prompt

System: You are an expert at analyzing peer review feedback.
Instructions: Given a review, your task is to classify each review comment according to two phenomena:
1. **Review Extension** – The feedback generates a full review instead of targeted feedback for a specific review segment. This includes writing additional evaluations, repeating comments, or elaborating beyond what is necessary.
2. **Role Separation** – The feedback provides suggestions meant for the author rather than the reviewer. Comments may directly address the author, suggest rewriting the review for the author, or cross the boundary between reviewing and authoring.
Task: - For each review comment, indicate whether it exhibits Review Extension, Role Separation, both, or neither. - Provide a brief justification for your classification.
Variables: - Review Comment: review
Output Format: - JSON array with fields:

```
[
  {
    "comment": "text of the review comment",
    "review_extension": true/false,
    "role_separation": true/false,
    "justification": "brief explanation"
  },
  ...
]
```

Figure 21: Prompt for identifying Review Extension and Role Separation in peer review comments.

Example Feedback: Extra Experiments

Issue Detected: The authors could also do [extra experiment X]
Reviewer Comment: "...no check for false negatives… only one language, one domain, one genre… no significance testing"
Feedback: Your comment highlights suggestions for additional experiments and concerns about generalization. While extra experiments (e.g., multilingual evaluation or statistical tests) are always possible, the paper provides sufficient evidence to support its main claims. The contribution introduces the DYNAMICER benchmark and a temporal segmented clustering method that improves retrieval-augmented generation for emerging mentions. These contributions are demonstrated convincingly within the chosen scope. Monolingual studies (English, sports domain, social media genre) are still valuable practically and theoretically. If you believe significance testing or cross-domain validation is essential for validity, please clearly justify why; otherwise, these points are better framed as suggestions rather than reasons for rejection.

Figure 22: Feedback generated from our approach on an instance of our dataset, LAZYREVIEWPLUS

Model	Variant	Const.	Relev.	Spec.	Conc.
Yi	Full Fitness Function	**3.9**	**3.8**	**3.8**	**3.8**
	w/o Length Score (sc_{len})	3.6	3.6	3.5	3.5
	w/o Template Adherence (sc_{temp})	3.7	3.7	3.6	3.6
	w/o Readability (sc_{read})	3.8	3.8	3.7	3.7
Qwen	Full Fitness Function	**3.8**	**3.8**	**3.7**	**3.7**
	w/o Length Score (sc_{len})	3.5	3.5	3.4	3.4
	w/o Template Adherence (sc_{temp})	3.6	3.6	3.5	3.5
	w/o Readability (sc_{read})	3.7	3.7	3.6	3.6
DeepSeek	Full Fitness Function	**3.5**	**3.6**	**3.6**	**3.5**
	w/o Length Score (sc_{len})	3.2	3.3	3.2	3.2
	w/o Template Adherence (sc_{temp})	3.3	3.4	3.3	3.3
	w/o Readability (sc_{read})	3.4	3.5	3.3	3.4
Oss.	Full Fitness Function	**4.0**	**4.1**	**4.0**	**3.8**
	w/o Length Score (sc_{len})	3.7	3.6	3.5	3.7
	w/o Template Adherence (sc_{temp})	3.8	3.7	3.6	3.7
	w/o Readability (sc_{read})	3.9	4.0	3.8	3.7
Phi	Full Fitness Function	**4.3**	**4.3**	**4.2**	**4.3**
	w/o Length Score (sc_{len})	3.8	3.7	3.8	3.6
	w/o Template Adherence (sc_{temp})	3.7	3.6	3.5	3.7
	w/o Readability (sc_{read})	4.1	4.0	4.0	4.1
	w/o Forbidden Term Penalty (pen_{forb})	4.2	4.2	4.1	4.3

Table 28: Ablation study on the components of the fitness function in the **feedback generation** approach across Constructiveness (Const.), Relevance (Relev.), Specificity (Spec.), and Conciseness (Conc.) for all models.

GPUs	Wall-clock Time (Best, h)	Wall-clock Time (Worst, h)
1	2.99	5.77
4	0.75	1.44
8	0.37	0.72

Table 29: Estimated wall-clock time for processing 5,000 reviews using multi-GPU parallelization.