Automated Refinement of Essay Scoring Rubrics for Language Models via Reflect-and-Revise

Keno Harada, Lui Yoshida, Takeshi Kojima, Yusuke Iwasawa, Yutaka Matsuo
The University of Tokyo
keno.harada@weblab.t.u-tokyo.ac.jp

Abstract

Large Language Models (LLMs) are increasingly used for Automated Essay Scoring (AES), yet the scoring rubrics they rely on are typically designed for human raters and may not be optimal for LLMs. Inspired by the calibration process that human raters undergo before formal scoring, we propose Reflect-and-Revise, an iterative framework that refines scoring rubrics by prompting models to reflect on their own chain-of-thought rationales and score discrepancies with human labels. At each iteration, the model identifies scoring-error patterns from sampled mismatches and revises the rubric accordingly. Experiments on three essay scoring benchmarks (ASAP, ASAP 2.0, and TOEFL11) with three LLMs (GPT-5 mini, Gemini 3 Flash, and Qwen3-Next-80B-A3B-Instruct) demonstrate that our method yields improvements in Quadratic Weighted Kappa (QWK), achieving gains of up to +0.403 over human-authored rubrics. Starting from a minimal seed rubric that specifies only the score scale, our method matches or exceeds expert rubric performance in most dataset-model combinations, indicating that iterative refinement can reduce the manual effort of rubric authoring. Analysis of the refined rubrics reveals that the refinement process introduces explicit procedural structures, such as conditional gating rules and quantitative thresholds, that are absent from human-authored rubrics, highlighting a gap between rubrics designed for human raters and those effective for LLMs.[1]

1 Introduction

Automated Essay Scoring (AES) systems powered by Large Language Models (LLMs) are increasingly expected to provide real-time, scalable feedback for students and alleviate the grading burden on instructors (Mizumoto and Eguchi, 2023; Yancey et al., 2023; Naismith et al., 2023; Pack et al., 2024). Typically, these systems employ static, pre-defined rubrics to guide the evaluation. However, it remains an open question whether rubrics designed for human raters are optimal for LLMs. When human raters use a rubric, they often engage in a collaborative calibration process: they score sample essays, discuss discrepancies in their judgments, and refine their shared understanding of the criteria to ensure consistency (Trace et al., 2016; Ozfidan and Mitchell, 2022; Yoo et al., 2025). This iterative, reflective practice is overlooked in current LLM-based AES, potentially limiting their alignment with human scoring patterns.

Recent studies show that LLMs have the ability to refine their own outputs especially when there is reliable external feedback (Madaan et al., 2023; Kamoi et al., 2024). Prompt optimization techniques leverage these capabilities to update prompts to maximize a targeted metric and show performance improvements in various tasks such as multi-hop reasoning, instruction following and privacy-aware delegation (Khattab et al., 2023; Opsahl-Ong et al., 2024; Agrawal et al., 2025).

Inspired by these developments and the calibration process of human raters, we propose an iterative Reflect-and-Revise approach for refining rubrics in LLM-based AES. Specifically, given a small set of 100 sample essays with human scores, the model iteratively refines the rubric by reflecting on its own scoring rationales and the discrepancies between its predicted scores and human labels, with the objective of maximizing Quadratic Weighted Kappa (QWK) between model and human scores.

Our method differs from prior rubric refinement approaches (Xie et al., 2024; Lee et al., 2024; Liu et al., 2024a) in two respects, both motivated by the improved reasoning and instruction-following capabilities of recent LLMs. First, we incorporate the model's chain-of-thought rationales into the revision process, providing richer diagnostic signals for identifying why the current rubric leads to

[1]The code is available at https://github.com/kenoharada/Reflect-and-Revise.

Study	Scores	Texts	Sources	Predictions	Rationales	Iterative
HD-Eval (Liu et al. (2024b))	✓					✓
MTS (Lee et al. (2024))			✓			
Xie et al. (2024)	✓	✓				
ActiveCritic (Xu et al. (2025))	✓	✓	✓			
AutoCalibrate (Liu et al. (2024a))	✓	✓	✓	✓		
Ours	✓	✓	✓	✓	✓	✓

Table 1: Comparison of signals and strategies used to refine evaluation rubrics. **Scores**: whether the method uses human-labeled scores. **Texts**: whether the method uses the texts being evaluated (essays/answers/responses that receive scores). **Sources**: whether the method uses source passages used to compose responses (e.g., source documents, provided essay themes). **Predictions**: whether the method uses model-predicted scores. **Rationales**: whether the method uses model-generated justifications (chain-of-thought rationales) accompanying predicted scores. **Iterative**: whether the method iteratively refines rubrics over multiple rounds. Our method uniquely leverages all five signals and performs iterative refinement.

Pattern	Example Snippet	Before	After	Δ
Conditional Gating	*. . . If it explores the "how" and "why" behind. . .*	0.0	21.2	+21.2
Quantitative Threshold	*. . . elaboration is limited to 2-3 sentences per point, it should . . .*	0.0	9.1	+9.1
Concrete Exemplification	*. . . if minor factual inaccuracies exist (e.g., misspelled names like. . .*	0.3	7.0	+6.7
Score Cap / Demotion	*. . . Do not award a 5 if the essay's analysis is. . .*	0.0	2.7	+2.7
Boundary / Tie-Break	*. . . - A 5 vs. 6 Distinction: A 6 must explore broader. . .*	0.0	2.6	+2.6
Stepwise Workflow	*. . . scoring algorithm (must be followed in order) Step 1. . .*	0.0	2.6	+2.6

Table 2: Regex-based match counts in human-authored rubrics (**Before**) vs. iteratively refined rubrics (**After**), averaged over three models and three datasets. Each pattern is detected by case-insensitive keyword matching. **Conditional Gating** (*if, when, unless, provided that*); **Quantitative Threshold** (*at least, at most, <=, >=, N reasons/examples/sentences, N%*); **Concrete Exemplification** (*e.g., for example, for instance*); **Score Cap / Demotion** (*cannot be Score, must not receive, do not award, downgrade, demotion*); **Boundary / Tie-Break** (*tie-break, borderline, threshold, N vs N, between adjacent*); **Stepwise Workflow** (*step N, checklist, workflow, procedure, in order*). Δ = After − Before. Detailed definitions, examples, and the regex counting procedure are provided in Appendix E.

scoring errors. Second, we apply iterative refinement over multiple rounds, enabling progressive rubric improvement rather than relying on a single revision pass. Table 1 summarizes the signals and strategies used for rubric refinement across prior work and our method.

We evaluate on three datasets (ASAP, ASAP 2.0, and TOEFL11) with three models (`GPT-5 mini`, `Qwen3-Next-80B-A3B-Instruct`, and `Gemini 3 Flash`). Our method consistently outperforms both human-authored rubrics and the AutoCalibrate baseline (Liu et al., 2024a) on ASAP and ASAP 2.0, achieving QWK gains of up to +0.403 over human rubrics. Starting from a minimal seed rubric that specifies only the score scale, our method matches or exceeds the performance of expert rubrics in most dataset–model combinations. Ablation studies show that the iterative refinement process leads to performance gains. Furthermore, analysis of the refined rubrics reveals that the refinement process introduces explicit procedural structures, such as conditional gating rules and quantitative thresholds, that are absent from human-authored rubrics.

2 Related Work

For non-verifiable tasks, where judging success is not as straightforward as in math or code, recent research has focused on LLM-based automatic evaluation using checklists and rubrics in prompts (Min et al., 2023; Qin et al., 2024; Lin et al., 2024; Wu

et al., 2025; Cook et al., 2025; Huang et al., 2025; Gunjal et al., 2025; Viswanathan et al., 2025; Lee et al., 2025; Xu et al., 2025; Liu et al., 2024a,b; Wen et al., 2025). AES is an example of such a non-verifiable task, and various techniques have been proposed (Mizumoto and Eguchi, 2023; Xie et al., 2024; Lee et al., 2024).

2.1 Rubric Design for LLM Evaluation

Recent studies suggest that the relationship between rubric design and LLM evaluation quality is not straightforward. Yoshida (2025) found that making rubrics more detailed does not always lead to performance gains in AES: three out of four models maintained similar scoring accuracy with a simplified rubric, and one model even showed decreased performance with more detailed rubrics. Similarly, Furuhashi et al. (2025) identified "negative items," rubric components that are valid for human evaluators but do not improve LLM performance, and showed that removing such items can even boost accuracy. These findings suggest that there remains room to find rubric formulations better suited for LLMs.

2.2 LLM-based Rubric Refinement

A growing body of work explores methods for generating or refining evaluation rubrics to improve agreement with human scores (Liu et al., 2024b; Lee et al., 2024; Xie et al., 2024; Xu et al., 2025; Liu et al., 2024a). Some methods generate rubrics in a single pass without subsequent revision: for example, generating rubrics from source passages (Lee et al., 2024), from few input–score examples (Xu et al., 2025), or by rewriting existing rubrics using human-labeled scores and evaluated texts (Xie et al., 2024).

Among these, Liu et al. (2024a) proposed the pipeline closest to ours. Their method samples candidate evaluation criteria from human-scored data, scores a held-out set, and refines the best-performing criteria by analyzing error cases where model scores diverge from human labels. However, their approach differs from ours in two key respects: (1) the refinement loop is executed only once rather than iteratively, and (2) the refinement step does not incorporate the model's chain-of-thought rationales, limiting the diagnostic signal available for rubric revision.

Our method extends this line of work by enabling the LLM to reflect on its own scoring output, including its chain-of-thought rationales, to iteratively refine the rubric. By feeding back not only human-labeled and model-predicted scores but also the model's justifications for those predictions, our approach provides richer diagnostic information for identifying why the current rubric leads to scoring errors. This process effectively mimics the calibration sessions of human evaluators, who refine their interpretations and build shared understanding before formal scoring (Trace et al., 2016; Ozfidan and Mitchell, 2022; Yoo et al., 2025). Table 1 summarizes the signals used for rubric refinement across prior work and our method.

3 Iterative Rubric Refinement

Our method iteratively refines rubric text from score mismatches between LLM predictions and human labels, and the model's chain-of-thought rationales. We provide the full algorithm in Appendix A.

3.1 Preliminaries

Let $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$ be the training set, where x_i is an essay response and y_i is the corresponding human score. Rubric refinement starts from an initial seed rubric rubric_0 (from coarse to detailed variants) and iteratively updates it over T iterations, maintaining a candidate pool of at most K rubrics for each iteration.

Given a rubric rubric, the evaluator LLM scores each training essay x_i, producing a predicted score $\hat{y}_i$ and a chain-of-thought rationale z_i:

$$(\hat{y}_i,\ z_i) = \text{LLM}(\text{rubric},\ x_i). \tag{1}$$

We measure rubric quality by Quadratic Weighted Kappa (QWK) between the predicted scores $\hat{\mathbf{y}}_{\text{rubric}} = (\hat{y}_1, \ldots, \hat{y}_N)$ and the human scores $\mathbf{y} = (y_1, \ldots, y_N)$, and seek:

$$\text{rubric}_{\text{best}} = \arg\max_{\text{rubric}}\ \text{QWK}(\hat{\mathbf{y}}_{\text{rubric}},\ \mathbf{y}). \tag{2}$$

For brevity, we hereafter write $\text{QWK}(\text{rubric})$ to denote $\text{QWK}(\hat{\mathbf{y}}_{\text{rubric}}, \mathbf{y})$ on the training set. QWK is a standard metric in automated essay scoring for measuring agreement with human raters (Ke and Ng, 2019). It is an agreement metric for ordinal labels and penalizes larger score gaps more heavily than smaller ones:

$$\text{QWK} = 1 - \frac{\sum_{i,j} w_{ij} O_{ij}}{\sum_{i,j} w_{ij} E_{ij}}, \quad w_{ij} = \frac{(i-j)^2}{(L-1)^2}, \tag{3}$$

Dataset	LLM	Rubric written by	QWK	Accuracy	Spearman
ASAP	Gemini 3 Flash	Human	0.481	0.385	**0.635**
	Gemini 3 Flash	AutoCalibrate	0.579	0.564	0.564
	Gemini 3 Flash	**Ours**	0.613	0.631	0.634
	Gemini 3 Flash	**Ours from simplest**	**0.642**	**0.648**	0.626
	GPT-5 mini	Human	0.119	0.123	0.324
	GPT-5 mini	AutoCalibrate	0.487	0.421	0.461
	GPT-5 mini	**Ours**	**0.522**	**0.466**	**0.540**
	GPT-5 mini	**Ours from simplest**	0.405	0.413	0.376
	Qwen3-80B-A3B	Human	0.096	0.107	0.259
	Qwen3-80B-A3B	AutoCalibrate	0.253	0.287	0.448
	Qwen3-80B-A3B	**Ours**	**0.480**	**0.531**	**0.485**
	Qwen3-80B-A3B	**Ours from simplest**	0.237	0.358	0.284
ASAP 2.0	Gemini 3 Flash	Human	0.619	0.420	0.688
	Gemini 3 Flash	AutoCalibrate	0.655	0.475	0.714
	Gemini 3 Flash	**Ours**	**0.725**	**0.562**	**0.747**
	Gemini 3 Flash	**Ours from simplest**	0.632	0.496	0.662
	GPT-5 mini	Human	0.287	0.315	0.451
	GPT-5 mini	AutoCalibrate	0.282	0.319	0.447
	GPT-5 mini	**Ours**	**0.474**	**0.400**	**0.510**
	GPT-5 mini	**Ours from simplest**	0.412	0.337	0.438
	Qwen3-80B-A3B	Human	0.544	0.333	0.606
	Qwen3-80B-A3B	AutoCalibrate	0.560	0.352	0.620
	Qwen3-80B-A3B	**Ours**	**0.622**	**0.408**	**0.631**
	Qwen3-80B-A3B	**Ours from simplest**	0.553	0.379	0.599
TOEFL11	Gemini 3 Flash	Human	0.631	0.705	**0.664**
	Gemini 3 Flash	AutoCalibrate	0.628	0.697	0.657
	Gemini 3 Flash	**Ours**	**0.633**	**0.743**	0.642
	Gemini 3 Flash	**Ours from simplest**	0.604	0.719	0.611
	GPT-5 mini	Human	0.526	0.705	0.566
	GPT-5 mini	AutoCalibrate	**0.538**	**0.711**	**0.578**
	GPT-5 mini	**Ours**	0.506	0.692	0.533
	GPT-5 mini	**Ours from simplest**	0.526	0.693	0.540
	Qwen3-80B-A3B	Human	0.424	0.515	0.516
	Qwen3-80B-A3B	AutoCalibrate	**0.595**	0.650	**0.615**
	Qwen3-80B-A3B	**Ours**	0.572	0.672	0.586
	Qwen3-80B-A3B	**Ours from simplest**	0.587	**0.686**	0.592

Table 3: Zero-shot evaluation results on all datasets. Bold and underlined values indicate the best and second-best scores for each model within each dataset.

where O_{ij} and E_{ij} are observed and expected confusion counts, and L is the number of score levels. QWK ranges from -1 to 1 and higher QWK is better: 1 indicates perfect agreement, 0 indicates chance-level agreement, and -1 indicates complete disagreement.

3.2 Reflect-and-Revise

At iteration t, we maintain a candidate rubric pool $\mathcal{C}_{t-1}$ of size at most K. For each candidate rubric $\text{rubric} \in \mathcal{C}_{t-1}$, we first score all training samples and collect failed examples:

$$\mathcal{E}(\text{rubric}) = \{(x_i, y_i, \hat{y}_i, z_i) \mid \hat{y}_i \neq y_i\}. \quad (4)$$

Then, we run M Monte Carlo trials, i.e., rubric generations from randomly drawn sub-samples. Specifically, for each Monte Carlo trial $m \in \{1, \ldots, M\}$ and each batch size $b \in \mathcal{B}$, we draw a balanced subsample $\tilde{\mathcal{E}} \subset \mathcal{E}(\text{rubric})$ of size b, stratified across score levels so that each human-score

value is represented as equally as possible. The model then rewrites the rubric by reflecting on the sampled errors and their rationales:

$$\text{rubric}' = \textsc{ReviseRubric}(\text{rubric}, \tilde{\mathcal{E}}). \quad (5)$$

Concretely, the revision prompt presents the current rubric together with each error case in $\tilde{\mathcal{E}}$—including the essay text, the model's predicted score and rationale, and the human score—and asks the model to identify scoring-error patterns and propose targeted rubric modifications (see Appendix B for the full prompt templates).

3.3 Iterative Update

Let $\mathcal{N}_t$ be the union of all newly revised rubrics and the previous top candidates $\mathcal{C}_{t-1}$. Every rubric in $\mathcal{N}_t$ is re-evaluated on $\mathcal{D}_{\text{train}}$, and we retain the top K by QWK:

$$\mathcal{C}_t = \text{TopK}_{\text{rubric}\in\mathcal{N}_t}\ \text{QWK}(\text{rubric}). \quad (6)$$

The global best rubric $\text{rubric}_{\text{best}}$ is updated only when the best candidate of $\mathcal{C}_t$ improves over the previous best training QWK. We repeat this process for T iterations and return $\text{rubric}_{\text{best}}$.

4 Experiments

4.1 Datasets

We evaluate on three essay scoring benchmarks. The Automated Student Assessment Prize (ASAP) dataset (Hamner et al., 2012) consists of student essays from U.S. standardized tests; we use essay set 1 (P1), which contains persuasive essays scored on an integer scale from 1 to 6 by human raters. ASAP 2.0 (Crossley et al., 2025) is a corpus of source-based argumentative essays written by U.S. secondary students across seven prompts, scored on a 1–6 integer scale with consistent rubrics and accompanying source texts; we use the "Exploring Venus" subset in our experiments. The TOEFL11 corpus (Blanchard et al., 2013) contains English essays written by non-native speakers across eight essay prompts, labeled at three proficiency levels (high, medium, low). For each dataset, we use 100 training samples for rubric refinement and evaluate on held-out test samples. Following the previous work setting (Lee et al., 2024), for ASAP and ASAP 2.0, we randomly sample 10% of the essays as a held-out test set and sample 100 essays from the remaining essays as the training set for rubric refinement. For TOEFL11, we use the original training and test splits provided by the dataset. The expert and simplest seed rubrics of ASAP are provided in Appendix D.'' Because our iterative refinement procedure incurs computational cost from repeated LLM inference, we prioritize coverage across multiple benchmarks over evaluating multiple prompts or subsets within the same benchmark. For ASAP and ASAP 2.0, we conduct experiments on one subset: ASAP essay set 1 (P1) and the ASAP 2.0 "Exploring Venus" subset.

For TOEFL11, the original rubric defines five proficiency levels (scores 1–5), but the dataset labels use three levels (high, medium, low). We adopt the rubric descriptions for score 4 as *high* (mapped to score 3), score 3 as *medium* (mapped to score 2), and score 2 as *low* (mapped to score 1).

4.2 Experimental Setup

We compare our Reflect-and-Revise method against two baselines. The first is the **Human Rubric** baseline, which directly uses the original human-authored rubric without any refinement. The second is **AutoCalibrate**, an adapted version of AutoCalibrate (Liu et al., 2024a) initialized from the same seed rubric as our method. AutoCalibrate is the closest prior work to ours because it refines evaluation criteria using human-labeled scores, evaluated texts, source passages, and model-predicted scores, as summarized in Table 1. We also choose AutoCalibrate as our primary baseline because many rubric refinement methods discussed in Table 1 incorporate post-processing steps beyond rubric modification, such as score calibration or aggregation, making it difficult to isolate the effect of rubric refinement itself.

In the original AutoCalibrate pipeline, the initial evaluation criteria are generated by an LLM and then refined using score discrepancies. Our method can also be initialized from an LLM-generated rubric. However, in this work, we focus on the effect of rubric revision itself: given the same initial human-authored rubric, how much can each method improve it? Therefore, we initialize both AutoCalibrate and our method from the same human-authored rubric. This controlled setting isolates the effect of the revision process from the effect of the initial rubric, and enables a direct comparison between single-pass revision and our iterative revision using model-generated scoring rationales.

This design also supports our analysis of how refinement changes human-authored rubrics. By starting from the human-authored rubric, we can

examine what kinds of linguistic and structural changes are introduced during refinement, and how the resulting rubrics differ from rubrics designed for human raters.

We evaluate using three frontier LLMs accessed via the OpenRouter API (OpenRouter Inc, 2025): `GPT-5 mini` (OpenAI, 2025), `Gemini 3 Flash` (Google, 2025), and `Qwen3-Next-80B-A3B-Instruct` (Qwen-3-80B-A3B for short) (Qwen Team, 2025). We report QWK, accuracy, Spearman correlation between model predictions and human scores on the held-out test set in a zero-shot setting. We experiment with two seed rubric variants. The first is an *expert* rubric, the full human-authored rubric provided with the dataset, which tests whether iterative refinement can further improve an already well-crafted rubric. The second is a *simplest* rubric, a minimal instruction specifying only the score scale (e.g., "Based on the response's content, rate the response on a scale of 1 to 6."). This variant serves two purposes: it examines whether our method can reduce the manual effort required for rubric authoring, and it tests whether starting from a minimal seed allows the refinement to explore a broader search space than one constrained by human-designed rubric structures. Detailed hyperparameters, including model-specific generation parameters, are provided in Appendix C.

5 Experimental Results

5.1 Main Results

Results on ASAP, ASAP 2.0 and TOEFL11 are provided in Table 3 respectively. For ASAP and ASAP 2.0, our method outperforms both the human rubric and AutoCalibrate baselines across all three models, with QWK gains of up to +0.403 over the human rubric (GPT-5 mini on ASAP) and +0.227 over AutoCalibrate (Qwen3-80B-A3B on ASAP). For TOEFL11, our method demonstrates comparable performance to the baselines.

5.2 Refinement from Simplest Seed Rubric

A key practical question is whether rubric refinement can reduce the burden of manual rubric authoring for LLMs. To investigate this, we apply our method starting from a *simplest* seed rubric and compare the resulting test QWK against the human expert rubric. As shown in Table 4, our method achieves comparable or better performance than the human expert rubric in most settings, with QWK gains of up to +0.286 (GPT-5 mini on ASAP) over the human rubric. This suggests that iterative refinement can reduce the need for manual rubric authoring.

Dataset	LLM	Ours from simplest	Δ vs. expert rubric
ASAP	GPT-5 mini	0.405	+0.286
	Gemini 3 Flash	0.642	+0.162
	Qwen3-80B-A3B	0.237	+0.140
ASAP 2.0	GPT-5 mini	0.412	+0.125
	Gemini 3 Flash	0.632	+0.014
	Qwen3-80B-A3B	0.553	+0.009
TOEFL11	GPT-5 mini	0.526	+0.001
	Gemini 3 Flash	0.604	−0.027
	Qwen3-80B-A3B	0.587	+0.162

Table 4: QWK comparison: rubrics refined from the simplest seed ("Based on the response's content, rate the response on a scale of 1 to 6.") vs. the rubric written by human experts. Δ denotes QWK improvement over the baseline using human expert rubrics. Our method achieves comparable or better performance than the human expert rubric in most settings, demonstrating that iterative refinement can reduce the need for manual rubric authoring.

6 Analysis

Analysis of Refined Rubrics To understand how refinement changes rubric content, we analyze the refined rubrics compared to the original human expert rubrics. Table 5 presents a word-count-based comparison of the original and refined rubrics. The refined rubrics tend to be longer up to approximately 10 times the length of the original rubrics (GPT-5 mini on TOEFL11). The overlapping words ratio between the original and refined rubrics is about 20% on average, indicating that the refined rubrics often introduce new content. Especially for GPT-5 mini, overlapping words ratio is as low as 5.1% on ASAP and leads to a large QWK gain of +0.403 suggesting that the refinement process can discover rubric formulations that are different from human-authored rubrics and better suited for the model's reasoning and scoring process. On the other hand, GPT-5 mini's refined rubric on TOEFL11 has a low overlapping words ratio of 3.3% but does not improve over the human rubric, suggesting that not all modifications lead to performance gains and has potential for overfitting to the training set.

We investigate what kinds of content the refinement process adds to rubrics. Through man-

Dataset	Model	Seed	Refined	Δ Words	Overlap (%)	Δ QWK
ASAP	Gemini 3 Flash	352	920	+568	19.8	+0.132
ASAP	GPT-5 mini	352	1,696	+1,344	5.1	+0.403
ASAP	Qwen3-80B-A3B	352	1,925	+1,573	12.3	+0.383
ASAP 2.0	Gemini 3 Flash	688	624	-64	41.0	+0.107
ASAP 2.0	GPT-5 mini	688	1,632	+944	1.1	+0.187
ASAP 2.0	Qwen3-80B-A3B	688	1,758	+1,070	38.5	+0.078
TOEFL11	Gemini 3 Flash	250	430	+180	30.9	+0.002
TOEFL11	GPT-5 mini	250	2,384	+2,134	3.3	-0.019
TOEFL11	Qwen3-80B-A3B	250	637	+387	18.8	+0.148
Average		430	1,334	+904	19.0	+0.158

Table 5: Quantitative comparison between seed and refined rubrics. The *Seed* rubric is the human-authored rubric used as the starting point for optimization; *Refined* is the rubric after iterative refinement. Word counts are shown for each, with Δ *Words* indicating the change. *Overlap* is the percentage of refined rubric words that also appear in the seed rubric (order-preserving longest common subsequence). Δ QWK is the improvement in test-set Quadratic Weighted Kappa after refinement (Refined − Seed).

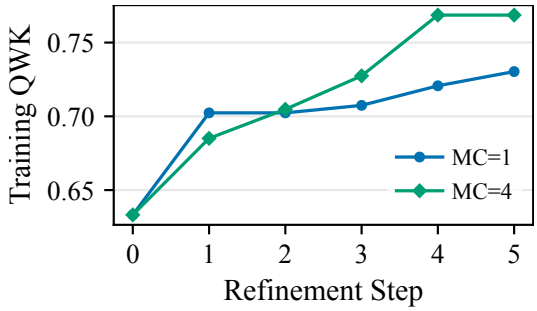

Figure 1: Best training QWK across iterations of Gemini 3 Flash on ASAP 2.0. MC stands for Monte Carlo trials. By using a large number of Monte Carlo trials, our method explores a wide variety of rubric modifications at each iteration, which leads to improvements in training QWK across iterations.

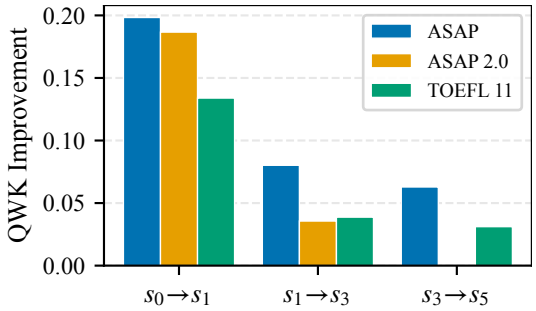

Figure 2: Best training QWK improvement between refinement steps, averaged across models for each dataset. $s_{t-n} \to s_t$ denotes the relative improvement from step $t-n$ to step t. The largest gains tend to occur in early iterations.

ual inspection of a sample of refined rubrics, we identified recurring linguistic and structural motifs and grouped them into six categories of rubric patterns: **Conditional Gating**, **Quantitative Threshold**, **Concrete Exemplification**, **Score Cap / Demotion**, **Boundary / Tie-Break**, and **Stepwise Workflow**. We then design regex-based detectors for each pattern and apply them to the original human rubrics and the refined rubrics, counting the number of matches for each pattern. The results are shown in Table 2. Detailed definitions and examples of these patterns are provided in Figure 4, and Figure 5 illustrates how the regex counts are computed in Appendix E.

The original human-authored rubrics contain almost none of these patterns. After refinement, all categories show increases. Conditional Gating exhibits the largest gain (+21.2), indicating that the refinement process frequently introduces explicit conditional rules to guide scoring decisions. Quantitative Threshold patterns (+9.1) show that the refined rubrics often specify numeric cutoffs to make scoring criteria more precise. Concrete Exemplification patterns suggest that the refined rubrics provide more specific examples to illustrate abstract criteria which are often missing in human rubrics. Figure 6 in Appendix E provides an example of a full refined rubric with detected patterns highlighted.

This pattern is consistent with prior work on improving human rater agreement. Jonsson and Svingby (2007) report that rubrics are more effective for aligning human judgments when they

LLM	Method	QWK
Gemini 3 Flash	Ours	**0.725**
Gemini 3 Flash	w/o Iteration ($T = 1$)	0.619
Gemini 3 Flash	w/o Rationale	0.717
GPT-5 mini	Ours	**0.474**
GPT-5 mini	w/o Iteration ($T = 1$)	0.285
GPT-5 mini	w/o Rationale	0.439
Qwen3-80B-A3B	Ours	**0.622**
Qwen3-80B-A3B	w/o Iteration ($T = 1$)	0.479
Qwen3-80B-A3B	w/o Rationale	0.531

Table 6: Ablation study on ASAP 2.0. The full method (Ours) outperforms both ablated variants, with larger drops observed when iterative refinement is removed than when chain-of-thought rationales are omitted.

Dataset	LLM	Baseline	$N = 20$	$N = 50$	$N = 100$
ASAP	Gemini 3 Flash	0.481	0.429	0.597	**0.613**
	GPT-5 mini	0.119	0.293	**0.529**	0.522
	Qwen3-80B-A3B	0.096	0.349	**0.508**	0.480
ASAP 2.0	Gemini 3 Flash	0.619	0.700	0.700	**0.725**
	GPT-5 mini	0.287	0.446	0.473	**0.474**
	Qwen3-80B-A3B	0.544	0.570	0.590	**0.622**
TOEFL11	Gemini 3 Flash	0.631	0.604	0.578	**0.633**
	GPT-5 mini	0.526	0.408	**0.543**	0.506
	Qwen3-80B-A3B	0.424	0.432	0.532	**0.572**

Table 7: Effect of training sample size (N) on test QWK. Baseline uses the human expert rubric without refinement. Bold indicates the best and underline indicates the second-best QWK among training sizes for each model.

provide topic-specific and concrete descriptions. The increases in Conditional Gating and Concrete Exemplification suggest that our refinement process moves rubrics in this direction by adding more explicit decision conditions and examples for interpreting abstract criteria. Prior work has also shown that rater training and score negotiation can reduce disagreement by helping evaluators develop shared interpretations of rubric categories (Trace et al., 2016). Our method mirrors this calibration process in an automated form: instead of human raters discussing discrepancies, the model uses its rationales and score mismatches with human labels to revise the rubric so that its future judgments better match human scoring patterns.

Iterative Refinement and Monte Carlo Effects To analyze the effect of iterative refinement and Monte Carlo trials, which control depth and breadth of exploration, respectively, we track the best QWK on the training dataset at each iteration. Figure 1 shows these trajectories of Gemini 3 Flash on ASAP 2.0. By using a large number of Monte Carlo trials, our method explores a wide variety of rubric modifications at each iteration, which leads to steady improvements in training QWK across iterations for all three models.

About gains from iteration, we calculate the stepwise improvement in QWK at each iteration, defined as the difference in best training QWK between steps. Figure 2 plots these stepwise improvements across iterations for all three datasets averaging across models. The largest gains tend to occur in early iterations, with diminishing returns in later iterations.

Ablation on Iterative and Rationale Components To quantify the contribution of iterative refinement and chain-of-thought rationales, we compare the full method with two ablated variants: (1) **w/o Iteration**, which performs only a single revision step ($T = 1$), and (2) **w/o Rationale**, which performs iterative refinement but omits the model-generated rationales from the revision prompt. Table 6 reports the results on ASAP 2.0, where the full method achieves the best QWK for all three models. Table 9 in Appendix F provides the full results across all three datasets. While removing iterative refinement causes the largest performance drop in most cases, for GPT-5 mini and Qwen3-80B-A3B on TOEFL11 and Gemini 3 Flash on ASAP, single-pass revision (w/o Iteration) achieved better performance than the full method. This suggests that while iterative refinement tends to improve performance, it may sometimes lead to overfitting. Developing further methods to utilize rationale information while preventing the overfitting caused by iterative refinement remains an important direction for future work.

Scoring Distribution Analysis To visualize how rubric refinement changes scoring behavior, Figure 3 shows confusion matrices for Qwen3-80B-A3B with the human expert rubric versus our refined rubric across all three datasets. After refinement, the model's predicted scores show higher agreement with human labels, as indicated by stronger diagonal patterns in the confusion matrices. Additional confusion matrices for GPT-5 mini and Gemini 3 Flash are provided in Appendix G.

Ablation on Training Data Size To analyze the effect of training data size on rubric refinement, we perform an ablation study using ASAP, ASAP 2.0, and TOEFL11 datasets. As shown in Table 7, we experiment with training data sizes of 20, 50, and 100 samples. Our method shows comparable or

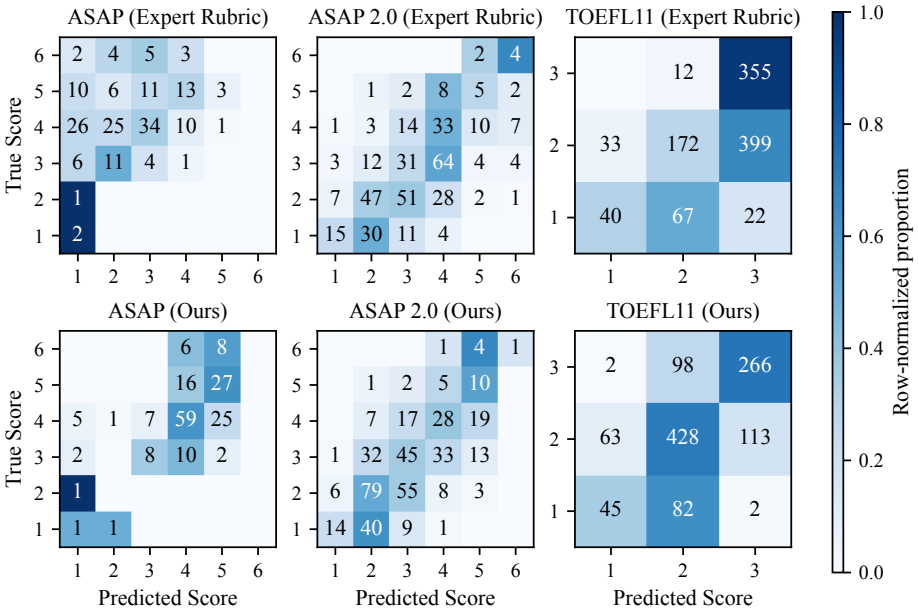

Figure 3: Confusion matrices for Qwen3-80B-A3B with the human expert rubric (top) and our refined rubric (bottom) across three datasets. Cell colors indicate row-normalized proportions; numbers show raw counts. Our refined rubric produces predictions more concentrated along the diagonal, indicating better agreement with human annotations.

better performance than the baseline even with as few as 20 or 50 training samples.

7 Conclusion

We proposed Reflect-and-Revise, an iterative framework that refines scoring rubrics for LLM-based Automated Essay Scoring by prompting models to reflect on their own chain-of-thought rationales and score discrepancies with human labels. Experiments on three benchmarks (ASAP, ASAP 2.0, and TOEFL11) with three LLMs demonstrated that our method improves scoring agreement with human raters, achieving QWK gains of up to +0.403 over human-authored rubrics. Starting from a minimal seed rubric that specifies only the score scale, our method matched or exceeded expert rubric performance in most dataset-model combinations, suggesting that iterative refinement can reduce the manual effort of rubric authoring for LLMs. Ablation studies showed that the iterative refinement process leads to larger performance gains, highlighting the importance of repeated revision based on observed scoring errors. Analysis of the refined rubrics revealed that the refinement process introduces explicit procedural structures, such as conditional gating rules and quantitative thresholds, that are absent from human-authored rubrics, highlighting a gap between rubrics designed for human raters and those effective for LLMs.

Limitations

Our work has several limitations. First, our analysis of refined rubric content relies on regex-based pattern matching, whose results depend on predefined pattern definitions. A more detailed analysis of the characteristics observed in refined rubrics would provide deeper insights into which rubric patterns are most effective for improving scoring performance. Second, our method requires a moderate number of human-labeled training samples (100 in our experiments) for rubric refinement. Although we showed that comparable gains are achievable with as few as 20 or 50 samples, the method still depends on the availability of human-scored data. Third, iterative refinement incurs computational cost due to repeated LLM inference over multiple iterations and Monte Carlo trials. For example, a single run of our method on ASAP 2.0 with Gemini 3 Flash cost approximately $170 in API calls. Finally, our evaluation is limited to English essay scoring across three benchmarks. Further evaluation across a broader range of models and datasets is needed to clarify when and where rubric refinement is preferable, including its generalizability to other languages, writing genres, and evaluation tasks.

References

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. 2025. Gepa: Reflective prompt evolution can outperform reinforcement learning. *Preprint*, arXiv:2507.19457.

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. Toefl11: A corpus of non-native english. *ETS Research Report Series*, 2013(2):i–15.

Jonathan Cook, Tim Rocktäschel, Jakob Nicolaus Foerster, Dennis Aumiller, and Alex Wang. 2025. TICKing all the boxes: Generated checklists improve LLM evaluation and generation.

Scott A Crossley, Perpetual Baffour, L Burleigh, and Jules King. 2025. A large-scale corpus for assessing source-based writing quality: ASAP 2.0. 65:100954.

Momoka Furuhashi, Kouta Nakayama, Takashi Kodama, and Saku Sugawara. 2025. Are checklists really useful for automatic evaluation of generative tasks? *Preprint*, arXiv:2508.15218.

Google. 2025. Gemini 3 flash: frontier intelligence built for speed.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. 2025. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *Preprint*, arXiv:2507.17746.

Ben Hamner, Jaison Morgan, lynnvandev, Mark Shermis, and Tom Vander Ark. 2012. The hewlett foundation: Automated essay scoring. `https://kaggle.com/competitions/asap-aes`. Kaggle.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, Xijun Gu, Peiyi Tu, Jiaxin Liu, Wenyu Chen, Yuzhuo Fu, Zhiting Fan, Yanmei Gu, Yuanyuan Wang, Zhengkai Yang, and 2 others. 2025. Reinforcement learning with rubric anchors. *Preprint*, arXiv:2508.12790.

Anders Jonsson and Gunilla Svingby. 2007. The use of scoring rubrics: Reliability, validity and educational consequences. 2:130–144.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. When can llms actually correct their own mistakes? a critical survey of self-correction of llms. *Transactions of the Association for Computational Linguistics*, 12:1417–1440.

Zixuan Ke and Vincent Ng. 2019. Automated essay scoring: A survey of the state of the art. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6300–6308. International Joint Conferences on Artificial Intelligence Organization.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *Preprint*, arXiv:2310.03714.

Sanwoo Lee, Yida Cai, Desong Meng, Ziyang Wang, and Yunfang Wu. 2024. Unleashing large language models' proficiency in zero-shot essay scoring. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 181–198, Miami, Florida, USA. Association for Computational Linguistics.

Yukyung Lee, Joonghoon Kim, Jaehee Kim, Hyowon Cho, Jaewook Kang, Pilsung Kang, and Najoung Kim. 2025. Checkeval: A reliable llm-as-a-judge framework for evaluating text generation using checklists. *Preprint*, arXiv:2403.18771.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2024. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. *Preprint*, arXiv:2406.04770.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024a. Calibrating LLM-based evaluator. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2638–2656, Torino, Italia. ELRA and ICCL.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024b. HD-eval: Aligning large language model evaluators through hierarchical criteria decomposition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7641–7660, Bangkok, Thailand. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc.

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

Atsushi Mizumoto and Masaki Eguchi. 2023. Exploring the potential of using an ai language model for automated essay scoring. *Research Methods in Applied Linguistics*, 2(2):100050.

Ben Naismith, Phoebe Mulcaire, and Jill Burstein. 2023. Automated evaluation of written discourse coherence using GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 394–403, Toronto, Canada. Association for Computational Linguistics.

OpenAI. 2025. Gpt-5 system card.

OpenRouter Inc. 2025. OpenRouter.

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. 2024. Optimizing instructions and demonstrations for multi-stage language model programs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9340–9366, Miami, Florida, USA. Association for Computational Linguistics.

Burhan Ozfidan and Connie Mitchell. 2022. Assessment of students' argumentative writing: A rubric development. *Journal of Ethnic and Cultural Studies*, 9(2):pp. 121–133.

Austin Pack, Alex Barrett, and Juan Escalante. 2024. Large language models and automated essay scoring of english language learner writing: Insights into validity and reliability. *Computers and Education: Artificial Intelligence*, 6:100234.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. InFoBench: Evaluating instruction following ability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13025–13048, Bangkok, Thailand. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3-next: Towards ultimate training & inference efficiency.

Jonathan Trace, Valerie Meier, and Gerriet Janssen. 2016. "i can see that": Developing shared rubric category interpretations through score negotiation. *Assessing Writing*, 30:32–43. Innovation in rubric use: Exploring different dimensions.

Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. 2025. Checklists are better than reward models for aligning language models. *Preprint*, arXiv:2507.18624.

Bosi Wen, Pei Ke, Yufei Sun, Cunxiang Wang, Xiaotao Gu, Jinfeng Zhou, Jie Tang, Hongning Wang, and Minlie Huang. 2025. HPSS: Heuristic prompting strategy search for LLM evaluators. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24974–25007, Vienna, Austria. Association for Computational Linguistics.

Xiaodong Wu, Minhao Wang, Yichen Liu, Xiaoming Shi, He Yan, Lu Xiangju, Junmin Zhu, and Wei Zhang. 2025. LIFBench: Evaluating the instruction following performance and stability of large language models in long-context scenarios. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16445–16468, Vienna, Austria. Association for Computational Linguistics.

Wenjing Xie, Juxin Niu, Chun Jason Xue, and Nan Guan. 2024. Grade like a human: Rethinking automated assessment with large language models. *Preprint*, arXiv:2405.19694.

Shuying Xu, Junjie Hu, and Ming Jiang. 2025. Large language models are active critics in nlg evaluation. *Preprint*, arXiv:2410.10724.

Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023. Rating short L2 essays on the CEFR scale with GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 576–584, Toronto, Canada. Association for Computational Linguistics.

Haneul Yoo, Jieun Han, So-Yeon Ahn, and Alice Oh. 2025. DREsS: Dataset for rubric-based essay scoring on EFL writing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13439–13454, Vienna, Austria. Association for Computational Linguistics.

Lui Yoshida. 2025. Do we need a detailed rubric for automated essay scoring using large language models? In *Artificial Intelligence in Education: 26th International Conference, AIED 2025, Palermo, Italy, July 22–26, 2025, Proceedings, Part VI*, page 60–67, Berlin, Heidelberg. Springer-Verlag.

A Algorithm

Algorithm 1 gives the full pseudocode for iterative rubric refinement.

BALANCEDSAMPLE$(\mathcal{E}, b)$ partitions the error set $\mathcal{E}$ into buckets by human-score level, then draws $\lfloor b/L' \rfloor$ examples from each bucket (where L' is the number of non-empty buckets), ensuring that each score level is represented as equally as possible.

B Prompt Templates

This section lists the exact prompt templates used in our pipeline. Placeholders enclosed in braces (e.g., `{rubric}`) are filled at runtime.

B.1 Evaluation Prompt

The following prompt is used to score each essay with the current rubric. The LLM is instructed to output a rationale followed by an integer score.

Algorithm 1 Iterative Rubric Refinement

Require: Training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$, seed rubric rubric_0, iterations T, pool size K, Monte Carlo trials M, batch sizes $\mathcal{B}$, evaluator LLM

Ensure: Best rubric $\text{rubric}_{\text{best}}$

$\mathcal{C}_0 \leftarrow \{\text{rubric}_0\}$; $\text{rubric}_{\text{best}} \leftarrow \text{rubric}_0$; $q_{\text{best}} \leftarrow \text{QWK}(\text{rubric}_0)$
for $t = 1$ **to** T **do**
  $\mathcal{N}_t \leftarrow \emptyset$
  for each rubric $\in \mathcal{C}_{t-1}$ **do**
    Score all $(x_i, y_i) \in \mathcal{D}_{\text{train}}$ with rubric $\rightarrow (\hat{y}_i, z_i)$
    $\mathcal{E}(\text{rubric}) \leftarrow \{(x_i, y_i, \hat{y}_i, z_i) \mid \hat{y}_i \neq y_i\}$
    for $m = 1$ **to** M **do**
      for each $b \in \mathcal{B}$ **do**
        $\tilde{\mathcal{E}} \leftarrow \textsc{BalancedSample}(\mathcal{E}(\text{rubric}),\, b)$
        $\text{rubric}' \leftarrow \textsc{ReviseRubric}(\text{rubric},\, \tilde{\mathcal{E}})$
        $\mathcal{N}_t \leftarrow \mathcal{N}_t \cup \{\text{rubric}'\}$
      end for
    end for
  end for
  $\mathcal{N}_t \leftarrow \mathcal{N}_t \cup \mathcal{C}_{t-1}$
  Re-evaluate all rubric $\in \mathcal{N}_t$ on $\mathcal{D}_{\text{train}}$
  $\mathcal{C}_t \leftarrow \text{TopK}_{\text{rubric} \in \mathcal{N}_t}\ \text{QWK}(\text{rubric})$
  if $\max_{\text{rubric} \in \mathcal{C}_t} \text{QWK}(\text{rubric}) > q_{\text{best}}$ **then**
    $\text{rubric}_{\text{best}} \leftarrow \arg\max_{\text{rubric} \in \mathcal{C}_t} \text{QWK}(\text{rubric})$; $q_{\text{best}} \leftarrow \text{QWK}(\text{rubric}_{\text{best}})$
  end if
end for
return $\text{rubric}_{\text{best}}$

Evaluation Prompt

You are an expert rater for student essays. Evaluate the essay strictly using the scoring guideline. Choose exactly one score from the scoring guideline's score points.

```
# Essay Prompt
"""{essay_prompt}"""
# Response
"""{response}"""
# Scoring Guideline
"""{rubric}"""
# Output format (follow exactly)
Rationale: [«<Brief evidence-based rationale.»>]
Rating: [«<One integer score only.»>]
```

B.2 Error-Case Format for Rubric Revision

For each incorrectly scored example fed into the revision prompt, the following format is used to present the error case to the model. When rationales are included (`with_rationale=True`), the model's chain-of-thought is shown alongside its predicted score.

Error-Case Format (with rationale)

```
Assistant input:
Essay prompt:
"""{essay_prompt}"""
Essay to be rated:
"""{response}"""
Assistant rationale:
"""{rationale}"""
Assistant score:
"""{rating}"""
Desired score:
"""{desired_rating}"""
```

B.3 Rubric Revision Prompt

The revision prompt wraps the current rubric and the sampled error cases, and instructs the model to output a revised rubric.

Rubric Revision Prompt (with rationale)

I asked an assistant to grade essays using the scoring guideline below:
“‘
{current_rubric}
“‘
Here are grading examples that include the assistant input, the assistant rationale, the assistant score, and

the desired score:
```
{examples}
```
Revise the scoring guideline to improve score agreement so the assistant's future ratings align more closely with the desired scores.
Requirements:
1. Use the rationale patterns to identify why the assistant over-scored or under-scored, and improve the scoring guideline guidance accordingly to reduce score mismatches.
Output rules:
– Return only the revised scoring guideline.
– Use exactly one fenced code block with triple backticks.
– Do not include any text before or after the code block.

When rationales are not used (`with_rationale=False`), the error-case format omits the `Assistant rationale` field, and the requirement in the revision prompt is replaced with: "Use score mismatch patterns to identify where the scoring guideline guidance is insufficient or ambiguous, and revise it to reduce score mismatches."

C Hyperparameters

Table 8 summarizes the hyperparameters. We adopt the number of Monte Carlo trials (M=4) from AutoCalibrate. For error batch sizes, AutoCalibrate originally used $\mathcal{B}$=$\{1, 2, 4\}$, but considering the improved capacity of recent LLMs to process longer contexts, we increase them to $\mathcal{B}$=$\{4, 8, 12\}$. For the number of iterations, the original self-refinement loop in Madaan et al. (2023) used T=4; we set T=5, expecting that the stronger reasoning and instruction-following capabilities of recent models would allow productive refinement over additional cycles. The AutoCalibrate baseline uses T=1 (single-pass revision) without rationales, while our Reflect-and-Revise method performs iterative refinement with rationale-based feedback. Both methods maintain K=3 top candidates and use N=100 training samples.

Hyperparameter	Ours	AutoCalibrate
Iterations (T)	5	1
Top-K candidates	3	3
Monte Carlo trials (M)	4	4
Batch sizes ($\mathcal{B}$)	{4, 8, 12}	{4, 8, 12}
Training samples (N)	100	100
With rationale	Yes	No

Table 8: Refinement hyperparameters for rubric refinement.

For all models, we set the maximum output tokens to 8192. For GPT-5 mini and Gemini 3 Flash, we set reasoning effort to low. All other generation parameters were left at their default values for each model.

Trying our method on ASAP 2.0 with Gemini 3 Flash cost approximately $170 (28 million input tokens and 50 million output tokens) in API calls, while GPT-5 mini and Qwen3-80B-A3B cost approximately $50 and $13, respectively. The exact costs depend on the number of tokens generated by the model, which can vary based on the length of the essays and the complexity of the revised rubrics.

D Seed Rubrics

This section lists the expert (human-authored) and simplest seed rubrics of the ASAP dataset used in our experiments. The expert rubric is the original human-authored rubric provided with the dataset, while the simplest rubric is a minimal description specifying only the score scale.

D.1 Expert rubric

Score Point 1: An undeveloped response that may take a position but offers no more than very minimal support. Typical elements: - Contains few or vague details. - Is awkward and fragmented. - May be difficult to read and understand. - May show no awareness of audience.
Score Point 2: An under-developed response that may or may not take a position. Typical elements: - Contains only general reasons with unelaborated and/or list-like details. - Shows little or no evidence of organization. - May be awkward and confused or simplistic. - May show little awareness of audience.
Score Point 3: A minimally-developed response that may take a position, but with inadequate support and details. Typical elements: - Has reasons with minimal elaboration and more general than specific details. - Shows some organization. - May be awkward in parts with few transitions. - Shows some awareness of audience.
Score Point 4: A somewhat-developed response that takes a position and provides adequate support. Typical elements: - Has adequately elaborated reasons with a mix of general and specific details. - Shows satisfactory organization. - May be somewhat fluent with some transitional language. - Shows adequate awareness of audience.
Score Point 5: A developed response that takes a clear position and provides reasonably persuasive support. Typical elements: - Has moderately well elaborated reasons with mostly specific details. - Exhibits generally strong organization. - May be moderately fluent with transitional language throughout. - May show a consistent awareness of audience.
Score Point 6: A well-developed response that takes a clear and thoughtful position and provides persuasive

support. Typical elements: - Has fully elaborated reasons with specific details. - Exhibits strong organization. - Is fluent and uses sophisticated transitional language. - May show a heightened awareness of audience.
Note: I have made an effort to remove personally identifying information from the essays using the Named Entity Recognizer (NER). The relevant entities are identified in the text and then replaced with a string such as "PERSON", "ORGANIZATION", "LOCATION", "DATE", "TIME", "MONEY", "PERCENT", "CAPS" (any capitalized word) and "NUM" (any digits). Please do not penalize the essay because of the anonymizations.

D.2 Simplest rubric

Based on the response's content, rate the response on a scale of 1 to 6.

E Analysis of Rubric Changes

This section provides analyses of the rubric changes introduced by our refinement method. Figure 4 defines the six structural patterns used in our regex-based rubric analysis, with representative matched snippets from refined rubrics. Figure 5 shows the regex counting procedure used to compare the initial and refined rubrics. Figure 6 shows a full refined rubric with detected patterns highlighted.

F Full Ablation Results

Table 9 reports the complete ablation results corresponding to the summary in Section 6. We evaluate two variants of Reflect-and-Revise against the full method. The **w/o Iteration** variant performs only a single revision step ($T = 1$), while the **w/o Rationale** variant keeps the iterative refinement procedure but removes model-generated rationales from the revision prompt. All variants are initialized from the same expert seed rubric and are evaluated on the same held-out test split as the main experiments.

G Additional Confusion Matrices

Figure 7 and Figure 8 provide the same scoring-distribution analysis as Figure 3 for GPT-5 mini and Gemini 3 Flash. In each figure, the top row uses the original human expert rubric, while the bottom row uses the rubric refined by our method. Columns correspond to ASAP, ASAP 2.0, and TOEFL11. Cell colors show row-normalized prediction proportions, and the overlaid numbers show raw counts.

Pattern Explanations with Random Matched Snippets

1. Conditional Gating
What this pattern captures: Condition-based branching rules (if / when / unless / provided that) that explicitly guide rater decisions under specific circumstances. Refined rubrics tend to add many conditional gates to reduce ambiguity in borderline situations.
Typical cues: if, when, unless, provided that
Example 1: "... .g., "@PERCENT1 of students," "@PERSON1 says...")-even **if** embedded in awkward phrasing or grammatical errors. - ..."
Example 2: "... istance between each grade should be considered equal. **When** scoring, prioritize the quality of critical thinking a ..."

2. Boundary / Tie-Break
What this pattern captures: Rules for resolving borderline cases between adjacent score bands. Includes tie-break procedures, explicit threshold cutoffs, and 'N vs N' comparisons (e.g., '3 vs 4'). Refined rubrics often add detailed boundary-resolution instructions to improve inter-rater agreement.
Typical cues: tie-break, borderline, threshold, N vs N, between adjacent
Example 1: "... chanics (E) first: B determines maximum possible band (**3 vs 4** vs 5/6), then apply E penalties. A and C then refine p ..."
Example 2: "... eaching 6 if the argument fails to cohere. 6. Holistic **tie-break** ers (final arbitration): - If features point to diff ..."

3. Stepwise Workflow
What this pattern captures: Ordered step-by-step procedures (Step 1, Step 2...) or checklists that structure the scoring process into a reproducible workflow. Optimization tends to transform free-form scoring guidance into structured, sequential procedures for raters to follow.
Typical cues: step N, checklist, workflow, procedure, in order
Example 1: "... e for tie-breaking and possible downward adjustment in **Step 4** . - If E = Minor: no reduction. Step 3 - Assess Cent ..."
Example 2: "... elopment. - Borderline handling (refined with explicit **checklist** and tie-breakers): When between adjacent scores, ask t ..."

4. Quantitative Threshold
What this pattern captures: Numeric cutoffs and quantified criteria (e.g., 'at least 2 facts', '~30% severe errors', '3 reasons') that replace vague qualitative descriptions with concrete numbers. Optimization frequently introduces numeric thresholds where the original rubric used imprecise terms like 'some' or 'several'.
Typical cues: at least, at most, <=, >=, N reasons/examples/sentences, N%
Example 1: "... and specific details/examples. Typical threshold: - **At least** 2 distinct reasons each with at least min-elab AND at ..."
Example 2: "... n. - Development depth requirement: expect roughly 2- **3 sentences** of elaboration per reason in a typical short essay (re ..."

5. Score Cap / Demotion
What this pattern captures: Hard constraints that cap the maximum achievable score or forcibly demote ratings when specific conditions are unmet. Examples: 'cannot receive 4 or higher', 'do not award 5', 'downgrade to 2'. Refined rubrics add these guards to prevent systematic over-scoring of essays that superficially appear competent.
Typical cues: cannot be Score, must not receive, do not award, downgrade, demotion
Example 1: "... id-idea and thereby leaves core reasoning undeveloped, **downgrade** one band. - Exceptions: If an essay otherwise meets ..."
Example 2: "... Score 4-even with numerous grammatical errors. Do not **downgrade** for language if the argument's logic and evidence are ..."

6. Concrete Exemplification
What this pattern captures: Detects rubric text that uses illustrative examples (e.g., for example, for instance) to clarify scoring criteria. Refined rubrics frequently replace abstract descriptions with example-rich explanations, making this a strong indicator of rubric practicality improvement.
Typical cues: e.g., for example, for instance
Example 1: "... dominate the text; some attempt at audience awareness (**e.g.** , letter format) is present but ineffective. - Ideas ar ..."
Example 2: "... te (briefly) which specific requirements were missing (**e.g.** , "only one reason developed; second reason absent," "e ..."

Figure 4: Overview of rubric-refinement patterns and representative rubric snippets. For each pattern, we provide a short interpretation and randomly sampled matched spans from refined rubrics; highlighted words indicate the cue expressions that triggered each pattern.

Regex counting procedure

```
import re
from pathlib import Path
PATTERNS = {
  "Conditional Gating":
    r"\bif \b |\bwhen \b |\bunless \b |\bprovided that\b ",
  "Quantitative Threshold": (
    r"at least|at most|<=|>=|\u 2264|\u 2265"
    r"|\b \d +\s *(?:reasons?|examples?|sentences?|words?|points?|facts?)\b "
    r"|\d +%|~\d +"
  ),
  "Concrete Exemplification":
    r"e\.g\.|for example|for instance",
  "Score Cap / Demotion": (
    r"cannot (?:be [Ss]core|receive|assign)|must not receive|"
    r"do not award|\bdowngrade \b |\bdemotion \b "
  ),
  "Boundary / Tie-Break": (
    r"tie-?break|borderline|\bthreshold \b |"
    r"\d \s *vs\.?\s *\d |between\s +adjacent"
  ),
  "Stepwise Workflow":
    r"\bstep \s +\d |checklist|workflow|procedure|\bin order\b ",
}
def count_matches(text: str, regex: str) -> int:
  return len(re.findall(regex, text, flags=re.IGNORECASE))
initial_text = Path("initial_rubric.txt").read_text(encoding="utf-8")
refined_text = Path("best_rubric.txt").read_text(encoding="utf-8")
for name, regex in PATTERNS.items():
  before = count_matches(initial_text, regex)
  after = count_matches(refined_text, regex)
  print(name, before, after, after - before)
```

Figure 5: Python-style code snippet of the regex-based counting procedure used for Table 2. For each pattern, the procedure counts case-insensitive regex matches in the initial rubric and the best refined rubric, then records the before count, after count, and their difference.

Pattern Legend

Conditional Gating (n=9)
Boundary / Tie-Break (n=2)
Stepwise Workflow (n=1)
Quantitative Threshold (n=2)
Concrete Exemplification (n=13)

Refined Rubric (Pattern-Highlighted)

Score Point 1: An undeveloped response that may take a position but offers no more than very minimal support. Typical elements:
- Contains only one or two extremely vague sentences.
- Is so fragmented that it is nearly impossible to discern a stance.
- Shows no awareness of the specific prompt or audience.

Note: If a response provides any discernible reasons or specific details, even with severe mechanical errors, move to at least a Score Point 2.

Score Point 2: An under-developed response that takes a position but provides little support. Typical elements:
- Contains only one general reason or a very short, unelaborated list of ideas.
- Lists items (e.g., "play games, go on Facebook") without any explanation of why they are good or how they work.
- Shows little or no evidence of structured organization.
- Highly simplistic or redundant language.
- Limited attempt to address the reader or the specific prompt requirements.

Score Point 3: A minimally-developed response that takes a position and provides a rudimentary level of support. Typical elements:
- Provides a list of reasons with very brief elaboration (usually only one sentence per point).
- Offers more general than specific details. Even if proper nouns or specific platforms (e.g., "Facebook," "Colombia") are mentioned, if the sentence following them provides no further development of the idea, the response remains a 3.
- Basic "listing" organization (e.g., "First, Second, Finally") with little internal development.
- Characterized by highly simplistic sentence structures and vocabulary.
- Contains frequent errors that may occasionally impede meaning.

Score Point 4: A somewhat-developed response that takes a position and provides adequate support. Typical elements:
- Addresses several reasons with a mix of general and specific details.
- Support often feels formulaic or relies heavily on the ideas/phrasing provided in the prompt (e.g., simply expanding on "exercise" and "nature" without unique hypothetical scenarios).
- Contains anecdotes or examples that are present but lack multi-layered development or "mechanics" (e.g., mentioning a surgery simulation or a personal fall but not exploring the broader implications).
- Shows satisfactory organization with a clear intro, body, and conclusion.
- May contain significant mechanical, spelling, or syntax errors; however, the argument is logically coherent.
- Language is functional but lacks fluency. Even if the essay introduces an original third point (like safety or jobs), if the prose is clumsy and the elaboration is limited to 2-3 sentences per point, it should remain a 4.

Score Point 5: A developed response that takes a clear position and provides moderately persuasive support. Typical elements:
- Has well-elaborated reasons with specific, original details that explore "mechanics" (e.g., describing the physical sensation of "getting fat," or the specific way a webcam creates "human interaction" compared to a phone).
- Moves beyond the prompt's suggestions by significantly expanding on how technology functions in personal or professional lives (e.g., how a nurse uses videos for procedures or the emotional impact of staying in touch with a friend who moved).
- Demonstrates a consistent persuasive tone and a clear attempt to engage the reader personally.
- While it may use a formulaic structure (e.g., "My first reason... My last reason..."), the depth of the elaboration and the use of original scenarios justify the 5. Depth of content overrides repetitive transitions at this level.
- Addresses the counter-argument or looks at the issue from multiple perspectives.

Score Point 6: A well-developed response that takes a clear and thoughtful position and provides persuasive, in-depth support. Typical elements:
- Fully elaborated reasons with numerous specific, concrete details or "multi-layered" anecdotes.
- Goes significantly beyond explaining "why" to explore the "implications" and "consequences" of the position (e.g., connecting computer use to global warming via power plants, or the specific danger of losing survival skills like lighting matches during a natural disaster).
- Exhibits sophisticated organization or creative framing (e.g., using a powerful opening quote or a rhetorical "hook").
- Shows a heightened awareness of audience through a compelling persuasive voice or a strong emotional hook that connects with the reader's daily life.
- Fluency is high; while minor mechanical errors may exist, the rhetorical variety, complexity of thought, and "voice" are strong enough to carry authority.

Note on Scoring:
- Evaluators must prioritize the depth and specificity of content over technical accuracy.
- "In-depth development" (Score 5 and 6) is defined by the student's ability to visualize a scenario for the reader or explain a cause-and-effect chain.
- A 4 vs. 5 Distinction: If an essay provides specific examples (like surgery simulations or medical stats) but fails to explain the *ripple effects* or *human impact* of those examples, it is a 4. If it explores the "how" and "why" behind the examples (e.g., the emotional relief of a webcam or the specific process of getting fit), it is a 5.
- A 5 vs. 6 Distinction: A 6 must explore broader societal or philosophical "implications" (e.g., global warming, natural disaster survival, the future of the environment) or use highly creative framing and sophisticated voice.
- Heavy repetition in sentence structure (e.g., "One reason is... Another reason is...") caps an essay at 4 ONLY IF the content is also basic/formulaic. If the content within those paragraphs is deep and explores implications, it should be moved to a 5.
- Proper nouns or platform names do not automatically grant a higher score; they must be accompanied by an explanation of *how* or *why* that specific item supports the argument.
- Anonymization markers (like @CAPS, @LOCATION) should be treated as the intended words/details.

Figure 6: Refined rubric of Gemini 3 Flash on ASAP. Background colors mark text spans in the refined rubric that match each pattern. The legend above shows the pattern-to-color mapping and match counts. When multiple patterns overlap on the same span, only one highlight is retained to keep the visualization readable. The rubric panel shows the full refined rubric.

Dataset	LLM	Method	QWK
ASAP	Gemini 3 Flash	Ours	0.613
	Gemini 3 Flash	w/o Iteration ($T = 1$)	**0.682**
	Gemini 3 Flash	w/o Rationale	<u>0.648</u>
	GPT-5 mini	Ours	**0.522**
	GPT-5 mini	w/o Iteration ($T = 1$)	0.360
	GPT-5 mini	w/o Rationale	<u>0.480</u>
	Qwen3-80B-A3B	Ours	<u>0.480</u>
	Qwen3-80B-A3B	w/o Iteration ($T = 1$)	0.369
	Qwen3-80B-A3B	w/o Rationale	**0.521**
ASAP 2.0	Gemini 3 Flash	Ours	**0.725**
	Gemini 3 Flash	w/o Iteration ($T = 1$)	0.619
	Gemini 3 Flash	w/o Rationale	<u>0.717</u>
	GPT-5 mini	Ours	**0.474**
	GPT-5 mini	w/o Iteration ($T = 1$)	0.285
	GPT-5 mini	w/o Rationale	<u>0.439</u>
	Qwen3-80B-A3B	Ours	**0.622**
	Qwen3-80B-A3B	w/o Iteration ($T = 1$)	0.479
	Qwen3-80B-A3B	w/o Rationale	<u>0.531</u>
TOEFL11	Gemini 3 Flash	Ours	<u>0.633</u>
	Gemini 3 Flash	w/o Iteration ($T = 1$)	0.535
	Gemini 3 Flash	w/o Rationale	**0.638**
	GPT-5 mini	Ours	0.506
	GPT-5 mini	w/o Iteration ($T = 1$)	<u>0.536</u>
	GPT-5 mini	w/o Rationale	**0.536**
	Qwen3-80B-A3B	Ours	0.572
	Qwen3-80B-A3B	w/o Iteration ($T = 1$)	**0.589**
	Qwen3-80B-A3B	w/o Rationale	<u>0.578</u>

Table 9: Full ablation results across all datasets and models. **Ours** denotes the full Reflect-and-Revise method with iterative refinement and model-generated rationales. **w/o Iteration** performs a single revision step ($T = 1$), and **w/o Rationale** removes rationales from the revision prompt while retaining iterative refinement. Bold and underlined values indicate the best and second-best QWK for each dataset–model combination.

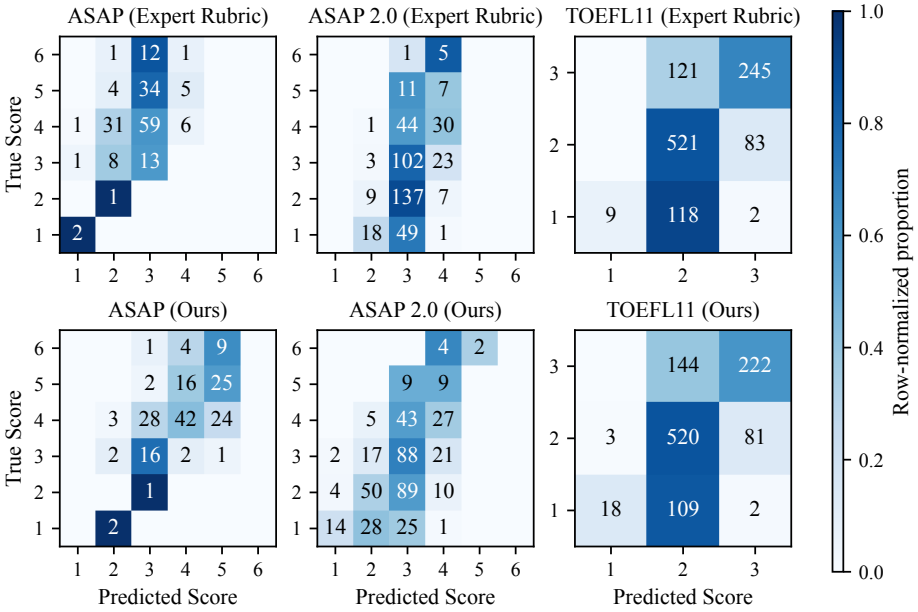

Figure 7: Confusion matrices for GPT-5 mini with the human expert rubric (top) and our refined rubric (bottom) across three datasets. Cell colors indicate row-normalized proportions; numbers show raw counts.

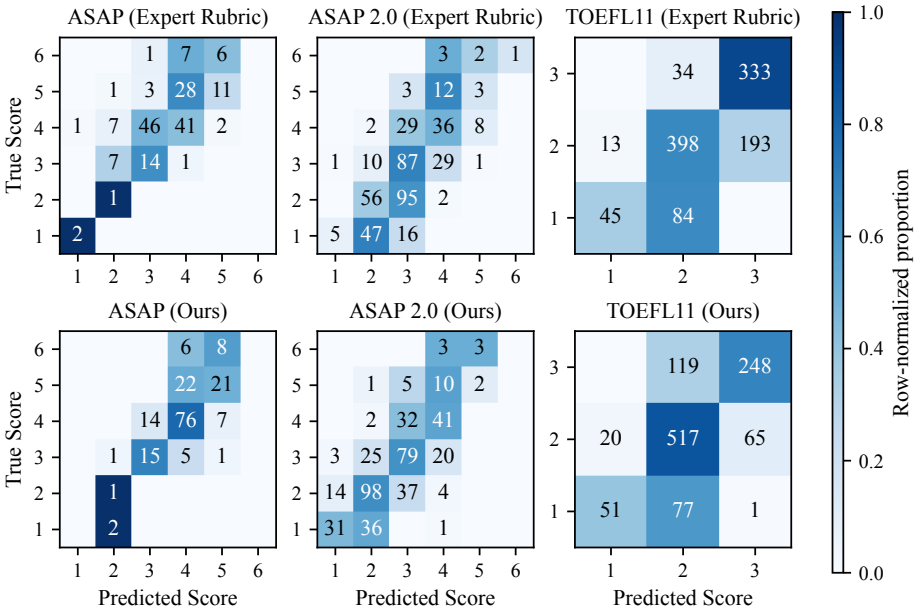

Figure 8: Confusion matrices for Gemini 3 Flash with the human expert rubric (top) and our refined rubric (bottom) across three datasets. Cell colors indicate row-normalized proportions; numbers show raw counts.