

LLM-Based Rubric Refinement in Multi-Agent Automated Scoring

Alexander Kwako and Cristina Everett and Harry Wang

Cambium Assessment, Inc.

alexander.kwako@cambiumassessment.com

Abstract

In this study, we marry automated scoring with automated rubric revision in an iterative, mutually reinforcing process. One LLM agent scores student responses; a second, the Error Analysis Agent, examines human–engine scoring discrepancies and proposes revisions to item-specific scoring rubrics. Revisions are kept only when they improve human–engine agreement. Based on the ASAP-SAS short constructed response dataset, results show that performance approaches a fine-tuned BERT baseline using only a fraction of the training data. A case study suggests that the revised rubrics elaborate on the original rubrics, which may be useful in the item development process.

1 Introduction

Large language models (LLMs) have opened new possibilities in educational assessment and, more specifically, in the fields of automated scoring and content generation (Ormerod et al., 2026). Agentic LLMs increase models’ capacity for processing and discerning patterns in large quantities of textual data, enabling them to tackle more complicated tasks.

Researchers have explored the possibility of using LLMs to not only produce a score, but also to generate a score rationale and feedback for students (Ormerod and Kwako, 2024; Stahl et al., 2024). Another possibility, which has received far less attention, is using LLMs to combine automated scoring with content generation for constructed response items. In this study, we demonstrate how human-scored responses can be used to improve automated scoring while simultaneously refining scoring rubrics. Our system employs multiple LLM agents—one to score, and another to revise the rubric—each interacting with the other to improve their output. Further, we provide a case study showing how the revised rubric elaborates on the original

rubric. Awareness of these gaps in the rubric, revealed through the rubric refinement process, may benefit item development.

1.1 LLM Agents

LLM agents (i.e., agentic AI) refer LLMs that are trained to perform multi-hop reasoning and use “tools” given large context windows (Yao et al., 2022). They constitute a meaningful step beyond simple instruction-tuned LLMs, having undergone additional rounds of training on highly curated data and through reinforcement learning (Shao et al., 2024). LLM agents can process large amounts of data, reason in sophisticated ways across a variety of problems, and consistently generate complex, structured output. As such, they are much better suited for sophisticated tasks like rubric refinement.

1.2 Constructed Response Items and Scoring Rubrics

Constructed response items are open-ended items, typically scored by human raters according to a scoring rubric and other supporting documentation (Haladyna et al., 2013). Scoring rubrics (and, sometimes, exemplar responses) are originally crafted by content experts during the writing of constructed response items. The scoring rubric is a key document that outlines how human raters should score a constructed response item. Rubrics typically provide a set of rules, definitions, and/or descriptions of all possible score points. Rubrics are often accompanied by exemplar responses that illustrate each score point; each response is paired with a brief rationale explaining the score it received.

After items are field tested, range finding is conducted in order to delineate the boundaries of each score point (Johnson et al., 2008). This process focuses on edge cases in particular, and decisions are made regarding what, for example, distinguishes a “high 1” from a “low 2”. During this process, the rubric may be amended and edge cases are doc-

umented. These may be used to help train future raters and document scoring changes.

The four documents (i.e., the scoring rubric, exemplar responses, rubric amendments, and edge cases) encapsulate the set of instructions that encapsulate how constructed responses items should be scored. Yet rubrics and supporting documents are often incomplete (Bejar, 2012). There sometimes remain tacit scoring rules and interpretations that determine how human raters score. Additional scoring rules and edge cases may be discussed but not formally documented. Parts of the rubric may be ambiguous, outdated, or otherwise in conflict with observed rater behavior. Potential gaps in the rubric may contribute to suboptimal inter-rater reliability (IRR).

1.3 Overview of Study

In this study, we combine an LLM scoring agent with an Error Analysis Agent (EAA) that refines the rubric used to score students’ responses. Optimizing scoring performance occurs in tandem with refining the scoring rubric. Given a batch of discrepancies between human raters and the scoring agent, the EAA is instructed to identify gaps in the scoring rubric—underspecifications, ambiguities, or inconsistencies—and to propose edits to mitigate them. Proposed edits are accepted or rejected depending on whether they improve human-engine agreement, based on quadratic weighted kappa (QWK). The design of our system is closely related to our two research questions:

1. Does rubric refinement improve automated scoring performance?
2. Does rubric refinement address legitimate gaps in the rubric?

We choose the term *rubric refinement* to emphasize the potential utility that such a system may offer to the item development process. Yet rubric refinement could also be conceptualized as *rubric inference* in the sense that the system is reverse-engineering the scoring rubric based on observed rater scores, or *prompt optimization* in that we aim to maximize automated scoring performance exclusively by altering the context of the scoring agent.

1.4 Related Literature

There are several other studies that integrate rubric refinement with automated scoring systems. Most similar to our own is Harada et al. (2026), who

describe a rubric refinement system that samples engine-human mismatches and refines rubrics over five iterations, selecting rubrics that maximize quadratic weighted kappa (QWK). They produce a sample of possible edits and evaluate each one, in contrast to our design which rejects or accepts single changes versioned with `git`. Additionally, they focus on essay scoring rather than short answer scoring.

Other studies describe similar systems, but these have larger variations in design or applications in different contexts. For instance, Chu et al. (2024) decompose error analysis into two separate steps: reflection and refinement. Wang et al. (2026) do not optimize on the rubrics directly, but rather on an “assessment skill” that generates the rubrics anew upon each iteration. Xie et al. (2025) describe a human-in-the-loop process designed to reduce hand-scoring labor; they use an LLM agent to draft the initial rubric, and later refine it using a sample of human scores and feedback.

2 Methods

2.1 Data

The Automated Student Assessment Prize: Short Answer Scoring (ASAP-SAS) dataset was originally made available to the public via a competition hosted by Kaggle in 2012 (Shermis, 2014). The ASAP-SAS dataset consists of 17,043 total responses across 10 items on the subject of science or English Language Arts (ELA)¹, administered to students in grades 8 and 10 (depending on the item). Each response was scored by two human raters. No resolution scores were provided; rater 1 scores serve as the final human score.

In order to maintain comparability with the extensive literature on this dataset, test-train splits were chosen to align with previous studies (e.g., Ormerod, 2022; Kumar and Boulanger, 2021). One of the advantages of using the ASAP-SAS dataset is that it is commonly used by other researchers, allowing us to compare our results with a wide range of previously established approaches. The size of the training and test sets for the ASAP-SAS dataset, in addition to some basic characteristics, are presented in Table 1.

¹The subjects “Biology” and “English” were renamed as “Science” and “ELA,” respectively, for consistency across items.

Item	Grade	Subject	N_{tot}	N_{test}	Average Word Count	H1H2 QWK	Score Point Distr.			
							0	1	2	3
1	10	Science	1672	335	47 (22)	.944	.23	.26	.31	.20
2	10	Science	1278	256	58 (23)	.914	.13	.26	.37	.25
3	10	ELA	1808	362	48 (15)	.755	.24	.56	.20	
4	10	ELA	1657	332	40 (16)	.749	.38	.54	.08	
5	10	Science	1795	359	26 (23)	.953	.77	.18	.02	.02
6	10	Science	1797	360	24 (22)	.956	.84	.09	.04	.03
7	10	ELA	1799	360	41 (25)	.968	.52	.25	.23	
8	10	ELA	1799	360	54 (34)	.856	.31	.26	.43	
9	10	ELA	1798	360	50 (37)	.828	.24	.41	.35	
10	8	Science	1640	328	41 (29)	.884	.18	.47	.35	

Table 1: ASAP-SAS items descriptive statistics. Note: Average Word Count includes standard deviation in parentheses. H1H2 QWK is human-human quadratic weighted kappa, based on the entire sample. Score Point Distributions are derived from H1 scores.

2.2 Rubrics and Other Documents

Accompanying the scored data are item-specific documents. Originally provided as item-specific Microsoft Word files, these were converted and broken into multiple, smaller Markdown files for each item:

1. metadata (i.e., item ID, response type, grade level, subject, score point range, and prompt title)
2. item content (i.e., stimuli and stem)
3. scoring rubrics

Formatting was preserved with some standardization. Word document headings, lists (e.g., bullet points), and tables were converted into their respective Markdown structures. Diagrams and figures (e.g., experimental apparatuses) were converted to PNG files and referenced in place with an XML `<figure>` placeholder carrying descriptive text. Data tables that appeared only as images were transcribed into Markdown tables rather than processed as images.

Expanded Documentation Experimental Condition. For the *expanded documentation* experimental condition, blank Markdown files were initialized for three additional documents. These were not part of the original data. Rather, they were created for this experiment to test whether an expansion and scaffolding of edits could improve performance. The additional, expanded set of documents include:

1. exemplars

2. rubric amendments

3. edge-cases

2.3 Models

Scoring Agent. For the scoring agent, we used Claude Haiku-4.5. The scoring agent is invoked frequently (once for every student response), so we chose a smaller, cheaper model for this task. As input, for the baseline rubric refinement condition, the scoring model takes in (1) metadata, (2) item content, (3) the scoring rubric, and (4) a single student response. It is instructed to output a JSON that includes (1) a rationale, grounded in the scoring rubric, and (2) a final score. The scoring agent prompt template for the baseline rubric refinement condition may be found in Appendix A.

Error Analysis Agent (EAA). For the EAA, we used Claude Opus-4.7. Input for the baseline rubric refinement condition included (1) metadata, (2) item content, (3) the scoring rubric, and (4) a discrepancy batch of student responses (i.e., randomly sampled human-engine scoring discrepancies, see Rubric Refinement Process for more details). The EAA is instructed to output a JSON that includes (1) a decision about whether to change the scoring rubric. If the EAA decides to make a change, it will also output (2) a description of the discrepancy it is addressing, (3) a description of the effect(s) the edit(s) are intended to have, (4) a list of response IDs used in making this decision, (5) the proposed changes, and (6) a git commit message summarizing the changes to the document(s). The EAA prompt template for the baseline rubric refinement condition may be found in Appendix B.

General Configurations. Both Haiku-4.5 and Opus-4.7 were accessed via the AWS Bedrock API. For Haiku-4.5, we used the default temperature of 1.0, since there is some evidence that non-zero temperature improves automated scoring performance (Frohn, 2026). Opus-4.7 does not include the option to vary temperature. With respect to max tokens, a maximum of 3,000 output tokens was budgeted for Haiku-4.5 and 20,000 was budgeted for Opus-4.7. These maximums were not hit when conducting the experiments.

Thinking Experimental Condition. For the baseline condition, thinking was not enabled for Haiku-4.5, but adaptive thinking was enabled for Opus-4.7. In the *thinking* condition, thinking was enabled for Haiku-4.5 with a budget of 2,000 tokens.

2.4 Rubric Refinement Process

At its core, the rubric refinement process is similar to the canonical machine learning algorithm: The process loops through a set of training data, optimizing an objective (i.e., QWK) within each cycle, until a prespecified number of cycles is completed. The primary difference is that, instead of updating model weights, it is the documentation—the same documentation that populates the scoring agent prompt template—that is updated. As the documentation improves, so does the scoring agent’s automated scoring performance. Figure 1 presents a flowchart of the rubric refinement process in greater detail.

Epoch 0. The rubric refinement process begins with Epoch 0, which accomplishes two tasks: (1) conducting the initial evaluation of the scoring agent (i.e., Claude-4.5) on the test set, and (2) generating the first discrepancy pool on the training set. The discrepancy pool is the set of student responses for which the scoring agent assigned a different score than the final human score. The discrepancy pool includes not only the scoring agent’s score, but its rationale and feedback (in the *feedback* condition only).

Improvement loops. The improvement loop begins by creating a git epoch branch (for the outer loop) and sub-epoch branch (for the inner loop). Within a sub-epoch, a sample of size D discrepancies is taken from the discrepancy pool and fed to the EAA along with the documentation. Based on the discrepancies, the EAA proposes edits to address any underspecifications, ambiguous instructions, or inconsistencies in the scoring rubric

or other documents (in the *expanded documentation* condition only). The entire discrepancy pool is reevaluated by the scoring agent and, if QWK improves, the sub-epoch branch is merged into the epoch branch and another discrepancy pool is generated. If QWK does not improve, the sub-epoch containing the proposed changes is discarded and the discrepancy pool remains unchanged. The inner loop runs J times before proceeding to the end of the outer loop. At the end of the outer loop, performance is reevaluated on the entire training set. If performance improves, as compared to the last full training set evaluation, then all changes in the epoch branch are merged into the main branch and another discrepancy pool is generated. If QWK does not improve, the epoch containing the proposed changes is discarded and the discrepancy pool remains unchanged. The outer loop runs I times before proceeding to the final evaluation on the test set.

Although the process would be simpler without an inner loop, it would also be slower and more expensive. The inner loop limits evaluation to the discrepancy pool, which is much smaller than the full training set.

Hyperparameter Selection. For this study, the following hyperparameters were chosen to balance cost and performance. Hyperparameter selection was also informed by preliminary runs, which showed that smaller J values tended to lead to fewer rejections, and that a larger sample size, N , could sustain a larger J and D . Performance tended to plateau quickly, within roughly 12 cycles, but this is subject to randomness and this number may be insufficient if a higher proportion of proposed edits are abandoned. No systematic hyperparameter optimization was conducted. Hyperparameters remained consistent across all experimental conditions: $N_{train} = 200$, $I = 6$, $J = 2$, $D = 20$.

2.5 Experimental Conditions

Five different sets of automated scoring results were generated for this study. These include:

1. **No refinement.** This condition captures the performance of the Claude-4.5 agent that relies on the original documentation only. Evaluation for this condition is conducted prior to rubric refinement.
2. **Rubric refinement.** This is the baseline rubric refinement condition, following the pro-

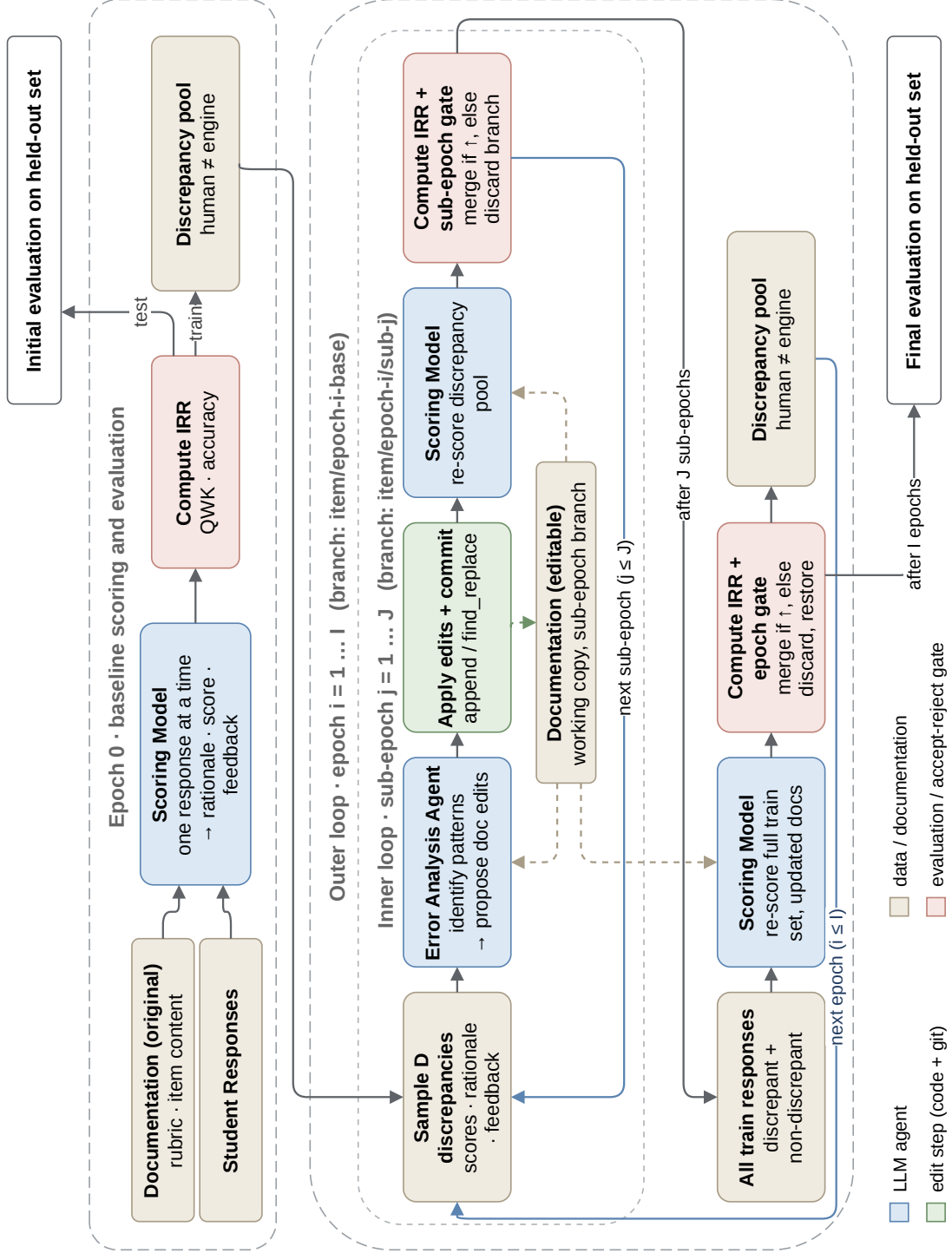


Figure 1: Rubric refinement process. Solid arrows show control flow; dashed arrows show documentation reads and writes.

cess described in the Rubric Refinement Process subsection above.

3. **Rubric refinement with thinking.** In this condition, thinking is enabled for the scoring agent. This is a model-level change; see the Models subsection above for details.
4. **Rubric refinement with feedback.** In this condition, the scoring agent is instructed to include feedback in its output, alongside a rationale and score. This is a process-level change; see the Rubric Refinement Process subsection above for details.
5. **Rubric refinement with expanded documentation.** Alongside the original scoring rubric, additional documents are initialized and editable by EAA throughout the process. This is a document-level change; see the Rubrics and Other Documents subsection above for details.

2.6 Performance Metric

We evaluate automated scoring performance based on Quadratic Weighted Kappa (QWK). This was the original metric specified in the Kaggle competition (Shermis, 2014, 2015). QWK quantifies the agreement between two raters beyond chance. It is a common metric for ordinal scaled items, penalizing disagreements more severely as the score point gap between raters widens. A QWK of 1 indicates exact agreement, whereas 0 indicates random agreement; negative values indicate worse than random disagreement. This metric is standard in the industry for comparing machine scoring performance (Williamson et al., 2012).

3 Results

3.1 Scoring Agent Performance

Rubric refinement improves automated scoring performance. Table 2 shows that the baseline QWK across the 10 ASAP-SAS items without rubric refinement is .517, using Claude 4.5 as the scoring model. With rubric refinement, however, the average QWK jumps to .716, nearly .2 higher.

In addition to rubric-only refinement, we tested three other experimental conditions that expand on the automated scoring system. Table 2 shows that two of the three additions improved QWK. The addition of thinking to the scoring model improved QWK by .024, a modest gain. Expanding documentation led to an improvement of .007, but this

is likely within the margin of error. By contrast, including feedback in model output decreased QWK by .016.

Although rubric refinement performed better than a comparable prompted proprietary model, e.g., GPT-4 (Jiang and Bosch, 2024), it did not perform better than traditional fine-tuning methods. BERT-base fine-tuning outperformed rubric refinement with thinking by .009, which is likely within the margin of error. LLM ensembling outperformed it by a substantial margin of .056.

3.2 Refined Rubric Case Study

As an example of how the scoring rubrics can change, we compared the original rubric of item 7 to the final rubric (edited in the baseline rubric refinement condition). Item 7 improved from .356 to .747, which is the largest gain among all the items. The original rubric is provided in Appendix C, and the refined rubric is provided in Appendix D.

The original scoring rubric for item 7 is specific and closely tied to the item stem. The stem asks students to “identify one trait that can describe Rose based on her conversations with Anna or Aunt Kolab. Include one detail from the story that supports your answer.” Correspondingly, the rubric requires that students (1) “identify one character trait,” and (2) “include story details that adequately support the answer.” The original rubric is relatively short.

The revised rubric leaves the language of the original rubric unchanged, and elaborates upon its meaning below it. The revision includes a new subheading, “Scoring Clarifications,” with two bolded subsections: (1) “What qualifies as a creditable character trait,” and (2) “What qualifies as an adequate supporting detail.” These map directly onto the two components of the item stem and original rubric.

The edits seem to address possible underspecifications or ambiguities in the original rubric, rather than address inconsistencies. In each of the bolded subsections, there is a long, bulleted list that elaborates on what counts as answering (or failing to answer) each part of the question. Bullet points include clarifications, exceptions, and many detailed examples. For instance, in one bullet point, the final rubric reads, “Temporary emotional or situational states do NOT count as character traits. Descriptions such as ‘tired,’ ‘stressed,’ ‘discouraged,’ ‘weighed down,’ ‘depressed,’ or ‘sad’ should receive Score 0 even if story details are provided.”

Scoring System	1	2	3	4	5	6	7	8	9	10	Avg
Human (Shermis, 2015)	.938	.911	.758	.686	.935	.973	.968	.837	.831	.904	.874
BERT-base (Ormerod and Kwako, 2024)	.849	.772	.692	.722	.845	.840	.676	.598	.829	.717	.749
LLM Ensemble (Ormerod, 2022)	.882	.891	.722	.750	.813	.822	.734	.702	.865	.779	.796
GPT-4 (Jiang and Bosch, 2024)	.715	.724	.626	.517	.772	.799	.495	.553	.703	.865	.677
Claude-4.5 (no refinement)	.610	.673	.371	.480	.653	.598	.356	.364	.444	.624	.517
Claude-4.5 (rubric refinement)	.843	.807	.634	.553	.768	.852	.747	.572	.697	.691	.716
+ thinking	.853	.860	.660	.606	.787	.878	.680	.670	.747	.656	.740
+ feedback	.840	.797	.512	.493	.773	.876	.696	.635	.716	.659	.700
+ expanded documentation	.856	.804	.600	.589	.797	.898	.584	.661	.765	.675	.723

Table 2: Results of rubric refinement across experimental conditions, including key benchmarks.

Although this addition may be implied in the term, “character trait,” which the original rubric uses, the revision makes clear that “character trait” does not include “temporary emotional or situational states.” Although this seems like a reasonable clarification, it is impossible to say whether this addition would have been aligned with the original intent of the question, or if the item writer would have phrased it in different conceptual terms.

Clarifications provided by the EAA may reveal additional ambiguities or potential inconsistencies. For example, the bullet point clarifying that temporary emotional states should not be credited is immediately followed by an addendum: “However, ‘worried for her family’ (or ‘concerned for her family’) IS creditable when the response ties the worry to Rose’s family rather than presenting it as a passing mood.” The EAA recognizes that some descriptions of Rose’s character could qualify as enduring or temporary depending on the context. It seeks to resolve this ambiguity that it, in part, created. Again, it is not clear whether this reflects the original intent of the rubric, or whether it was specified elsewhere during item development.

4 Discussion

4.1 Summary

The rubric refinement method offers a way to improve automated scoring performance and the scoring rubric in parallel. Although results are promising, the performance of the automated scoring system is not competitive with traditional fine-tuning approaches. Yet this method also requires far less data than fine-tuning approaches—200 labeled responses as compared to 1,300 for fine-tuning. By leveraging training data to update the scoring rubric, this method is particularly relevant for researchers and developers who do not have access to large amounts of data or whose systems depend on proprietary or closed-source models. This method

also offers benefits to content experts aiming to refine their own documentation during the development of constructed response items.

The fact that automated scoring performance improves alongside rubric refinement indicates that EAA changes are adding useful context for the automated scoring model. Information from the training data that is salient to the scoring agent is incorporated into the rubric. Whether EAA edits produce a rubric that is more comprehensive or more valuable to *human raters*, however, is a question that requires additional study. But preliminary evidence, based on the case study provided in the results section, suggests that these changes are relevant and may be valuable beyond the context of an automated scoring agent.

4.2 Limitations

The rubric refinement process described in this study depends on having a set of human-scored responses. Thus, this process would only be relevant after field-testing has already occurred. A separate process would have to be designed and evaluated if the goal were to write the initial draft of the rubric and exemplar responses.

In its current form, the rubric refinement process is slow and expensive. For these reasons, it is also difficult to optimize and iteratively improve the process. There are a number of potential enhancements to explore, from basic hyperparameter sweeps and increased training data to more sophisticated upgrades like adding git history to the EAA context, using a multi-agent scoring engine (e.g., Du et al., 2023), adding a rubric-editing step, or providing EAA with a more targeted sampling strategy when drawing from the discrepancy pool. Unfortunately, rigorously testing each of these is expensive or cost prohibitive.

This study also lacks an analysis of the refined rubrics and how the rubrics evolve throughout the refinement process. The quantitative evaluation

based on QWK is a heuristic for improving the rubric. This approach stops far short of validating the changes to the rubric. Ideally, this study would be conducted concurrent with item development such that content experts could evaluate the changes and verify that they remained aligned to the construct and/or standards that the items are designed to measure.

4.3 Next Steps

As mentioned above, there are numerous improvements that could be made to the rubric refinement process. In terms of optimizing automated scoring performance, one of the most high-leverage changes could be switching the scoring agent from Claude-4.5 to an open source model, such as Qwen3 (Yang et al., 2025). Moving to an open source model would not only decrease costs but also allow us to update model weights using supervised fine-tuning (Wei et al., 2021) or Group Relative Policy Optimization (Shao et al., 2024). This change may combine the best of rubric refinement with traditional training methods.

References

- Issac I. Bejar. 2012. [Rater Cognition: Implications for Validity](#). *Educational Measurement: Issues and Practice*, 31(3):2–9.
- Yucheng Chu, Hang Li, Kaiqi Yang, Harry Shomer, Hui Liu, Yasemin Copur-Gencturk, and Jiliang Tang. 2024. [A llm-powered automatic grading framework with human-level guidelines optimization](#). *arXiv preprint arXiv:2410.02165*.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. [Improving factuality and reasoning in language models through multiagent debate](#). *arXiv preprint arXiv:2305.14325*.
- Scott Frohn. 2026. [The Impact of LLM Self-Consistency and Reasoning Effort on Automated Scoring Accuracy and Cost](#). *arXiv preprint arXiv:2604.26954*.
- Thomas M. Haladyna, Michael C. Rodriguez, and Michael C. Rodriguez. 2013. [Developing and validating test items](#), volume 410. Routledge New York, NY.
- Keno Harada, Lui Yoshida, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. 2026. [Automated refinement of essay scoring rubrics for language models via reflect-and-revise](#). In *Proceedings of the 30th Conference on Computational Natural Language Learning*, pages 771–789.
- Lan Jiang and Nigel Bosch. 2024. [Short answer scoring with GPT-4](#).
- Robert L. Johnson, James A. Penny, and Belita Gordon. 2008. [Assessing performance: Designing, scoring, and validating performance tasks](#). Guilford Press.
- Vivekanandan S. Kumar and David Boulanger. 2021. [Automated Essay Scoring and the Deep Learning Black Box: How Are Rubric Scores Determined?](#) *International Journal of Artificial Intelligence in Education*, 31(3):538–584.
- Chris Ormerod and Alexander Kwako. 2024. [Automated Text Scoring in the Age of Generative AI for the GPU-poor](#). *Chinese/English Journal of Educational Measurement and Evaluation*, 5(3).
- Christopher Ormerod. 2022. [Short-answer scoring with ensembles of pretrained language models](#). *arXiv preprint*. ArXiv:2202.11558 [cs].
- Christopher Ormerod, Kai North, Alexander Kwako, and Susan Lottridge. 2026. Large Language Models in Assessment. In Xinhui Xiong, Mark Shermis, and Jiawei Xiong, editors, *The Role of AI in Assessment*, pages 205–234. Routledge, New York, NY.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models](#). *arXiv preprint*. ArXiv:2402.03300 [cs].
- Mark D. Shermis. 2014. [State-of-the-art automated essay scoring: Competition, results, and future directions from a United States demonstration](#). *Assessing Writing*, 20:53–76.
- Mark D. Shermis. 2015. [Contrasting State-of-the-Art in the Machine Scoring of Short-Form Constructed Responses](#). *Educational Assessment*, 20(1):46–65. [preprint: https://doi.org/10.1080/10627197.2015.997617](https://doi.org/10.1080/10627197.2015.997617).
- Maja Stahl, Leon Biermann, Andreas Nehring, and Henning Wachsmuth. 2024. [Exploring LLM Prompting Strategies for Joint Essay Scoring and Feedback Generation](#). *arXiv preprint*. ArXiv:2404.15845 [cs].
- Yun Wang, Xin Xia, Xuansheng Wu, Xiaoming Zhai, and Ninghao Liu. 2026. [Learnable Assessment Skills for LLM-based Automated Scoring: Rubric Construction via Iterative Optimization](#). *arXiv preprint arXiv:2605.29274*.
- Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. [Finetuned language models are zero-shot learners](#). *arXiv preprint arXiv:2109.01652*.
- David M. Williamson, Xiaoming Xi, and F. Jay Breyer. 2012. [A Framework for Evaluation and](#)

Use of Automated Scoring. *Educational Measurement: Issues and Practice*, 31(1):2–13. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1745-3992.2011.00223.x>.

Wenjing Xie, Juxin Niu, Chun Jason Xue, and Nan Guan. 2025. [Grade Like a Human: Rethinking Automated Assessment with Large Language Models](#). In *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, pages 1–8, Ho Chi Minh Vietnam. ACM.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 Technical Report](#). *arXiv preprint*. ArXiv:2505.09388 [cs].

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. [React: Synergizing reasoning and acting in language models](#). *arXiv preprint arXiv:2210.03629*.

A Scoring Agent Prompt Template

The scoring agent prompt template for the baseline rubric refinement condition is presented below.

```
## Overview of Task
You are an **educational assessment content expert** tasked with scoring a student's response to a
    ↪ constructed-response item using a scoring rubric.

Outputs you will provide (in JSON format, specified below):
1. Rationale
2. Score

Inputs you will receive:
1. Item Metadata
2. Item Content (including stimuli and task prompt)
3. Scoring Rubric
4. Student Response

## Output Specifications
- `rationale` is 3-5 sentences that:
    - cites concrete evidence from the student response,
    - references aspects of the scoring rubric relevant to the student response,
    - makes an argument for why the response should be assigned a particular score point.
- `score` is an integer in the allowed range for this item.

Additional Notes:
- Length of student response is not evidence; a short response containing the credited element(s)
    ↪ should earn the point(s).

### Required Output Format
Return **only valid JSON** in this exact schema:

```json
{
 "rationale": "Your rationale for giving the intended score.",
 "score": "integer score"
}
```

---

## Inputs

Item-specific documents, including the scoring rubric, are provided below to help you score the
    ↪ student response, which is provided last. Each document is wrapped in triple quotes (``').

### Item Metadata
'''
{{ITEM_METADATA}}
'''

### Item Content
'''
{{ITEM_CONTENT}}
'''

### Scoring Rubric
'''
{{RUBRIC}}
'''

### Student Response
'''
{{STUDENT_RESPONSE}}
'''
```

B EAA Prompt Template

The EAA prompt template for the baseline rubric refinement condition is presented below.

```
## Overview of Task
```

You are an **educational assessment content expert** tasked with improving a constructed-response scoring engine. Your objective is to improve human-engine agreement by revising a **scoring rubric** so it is clearer to the automated scoring engine how to score student responses. You should identify where documentation is underspecified, ambiguous, or inconsistent with observed rater behavior, then propose lean documentation updates.

Outputs you will provide (in JSON format, specified below):

1. Decision
2. Discrepancy Description
3. Expected Effect
4. Evidence Response IDs
5. Proposed Changes
6. Commit Message

Inputs you will receive:

1. Item Metadata
2. Item Content (includes stimuli and task prompt)
3. Scoring Rubric
4. Discrepancy Batch (includes response ID, response text, human score(s), engine score, and engine rationale)

Documentation Update Policy

`rubric.md` is the only document the scoring model is shown and the only one you may edit. Everything you propose – clarifications, tacit rules, borderline resolutions, exemplar responses – goes into it. Edit it when its wording is underspecified, ambiguous, or inconsistent with observed rater behavior.

If no room for improvement may be found, choose **"NO_CHANGES"** for your decision.

If a revision is warranted, choose **"PROPOSE_CHANGES"** for your decision, and use one of these

- operation types:
 - **"append"**: add to the end of a document without changing existing text. Provide the new content in a `"content"` field.
 - **"search_replace"**: make targeted edits to existing text. Provide a `"changes"` list where each entry has a `"find"` string (text to locate, must match exactly) and a `"replace"` string (the replacement text). Use this instead of rewriting entire documents.

Output Specifications

Return **only valid JSON**. Example JSON output (Option A and Option B) supplied below.

Option A: No change

```
```json
{
 "decision": "NO_CHANGES"
}
```
```

Option B: Propose changes

```
```json
{
 "decision": "PROPOSE_CHANGES",
 "discrepancy_description": "Recurring discrepancy pattern",
 "expected_effect": "How these edits should reduce specific discrepancy patterns.",
 "evidence_response_ids": ["id1", "id2", "id3"],
 "proposed_changes": {
 "rubric.md": {
 "operation": "search_replace",
 "changes": [
 {
 "find": "exact text to find in the document",
 "replace": "replacement text"
 }
]
 }
 },
 "commit_message": "[item-N]: Git commit message\n- rubric.md: ...\nRationale: ..."
}
```
```

Additional Notes

- This is an iterative procedure, so your goal shouldn't be to fix every single weakness in one pass.
- Keep updates concise; small changes are still legitimate changes to make.
- Focus on limitations that are easy to fix or patterns that spans multiple responses in the
 - ↳ Discrepancy Batch.
- Prefer updates when supported by multiple examples.
- Removing redundancy in the scoring rubric is a legitimate change to make, especially if the engine
 - ↳ seems to be ignoring instructions provided in the scoring rubric.
- Rephrasing or changing the emphasis of the scoring rubric are legitimate changes to make,
 - ↳ especially if the engine seems to be ignoring instructions provided in the scoring rubric.
- Although human scores are the gold standard, human raters sometimes make mistakes; try to
 - ↳ distinguish likely documentation gaps from likely human scoring errors.
- When documentation and observed rater behavior conflict, explicitly identify the conflict and
 - ↳ recommend the minimal corrective action.
- Draft the proposed change and test it against the batch before you commit. Check that it would not
 - ↳ flip responses the engine currently gets right.
- Verify your `find` strings. Every `search\_replace` entry is applied by exact string match and is
 - ↳ silently skipped if it does not match. Copy each `find` verbatim from the document text above
 - ↳ - do not reconstruct it from memory or normalize its whitespace, punctuation, or
 - ↳ capitalization.
- Examine the discrepancies not only by their score point differences, but by the **reason** the engine
 - ↳ and the human diverged, as different score point differences may be driven by similar
 - ↳ reasons.
- Response IDs should be included in the json output, but do not include specific IDs in the rubric,
 - ↳ as this will clutter the documentation, which should remain concise and clear for the scoring
 - ↳ engine.

Inputs

Item-specific documents, including the scoring rubric, are provided below. Each document is wrapped
 ↳ in triple quotes (``'). The Scoring Rubric is the one editable document that you may propose
 ↳ changes to. The Discrepancy Batch, provided at the end, includes the sample of human-engine
 ↳ scoring discrepancies which should form the crux of your analysis.

Item Metadata

```
'''
{{ITEM_METADATA}}
'''
```

Item Content

```
'''
{{ITEM_CONTENT}}
'''
```

Scoring Rubric (Editable)

```
'''
{{RUBRIC}}
'''
```

Discrepancy Batch

The following are human-engine scoring discrepancies for your analysis:

```
{{DISCREPANCY_BATCH}}
```

C Item 7 Rubric (Original)

The rubric of item 7 that was made publicly available is presented below. This text was reformatted into a Markdown document (see the Rubrics and Other Documents subsection for details).

Rubric for Trait of Rose

This item requires a two-part response. The student must identify one character trait of Rose, as
 ↳ well as include story details that adequately support the answer. The response should be
 ↳ related to the conversations with Anna or Aunt Kolab. In order to receive ANY credit, the
 ↳ student must provide a creditable response to the first part of the question. An additional
 ↳ point is awarded if the student provides a creditable response to the second part of the
 ↳ question.

Item-Specific 2-Point Rubric

- **Score 2:** The response identifies one character trait of Rose AND includes a story detail that
↳ adequately supports the answer, related to her conversations with Anna or Aunt Kolab.
- **Score 1:** The response identifies a character trait of Rose.
- **Score 0:** No points are given if a character trait of Rose's is not identified.

D Item 7 Rubric (Refined)

The rubric of item 7 following the baseline rubric refinement process is presented below.

Rubric for Trait of Rose

This item requires a two-part response. The student must identify one character trait of Rose, as
↳ well as include story details that adequately support the answer. The response should be
↳ related to the conversations with Anna or Aunt Kolab. In order to receive ANY credit, the
↳ student must provide a creditable response to the first part of the question. An additional
↳ point is awarded if the student provides a creditable response to the second part of the
↳ question.

Item-Specific 2-Point Rubric

- **Score 2:** The response identifies one character trait of Rose AND includes a story detail that
↳ adequately supports the answer, related to her conversations with Anna or Aunt Kolab.
- **Score 1:** The response identifies a character trait of Rose.
- **Score 0:** No points are given if a character trait of Rose's is not identified.

Scoring Clarifications

****What qualifies as a creditable character trait (required for ANY credit):****

- The trait must describe an enduring quality of Rose's character (e.g., caring, compassionate,
↳ supportive, selfless, responsible, hardworking, dutiful, worried for her family). This list
↳ is illustrative, not exhaustive. Other trait words are ALSO creditable when they name an
↳ enduring quality that is defensible from Rose's conversations with Anna or Aunt Kolab,
↳ including: understanding, sincere, mature, considerate, thoughtful, respectful, protective,
↳ nurturing, motherly, strong-willed, mentally strong, determined, and reassuring. Descriptive
↳ phrasing such as "acts like a mother," "mother figure," or "has traits like a mother would
↳ have" DOES count as explicitly naming a creditable trait (motherly/nurturing) so long as the
↳ response makes the enduring quality clear.
- The trait must be defensible from Rose's conversations with Anna or Aunt Kolab. A trait derived
↳ only from other parts of the story (e.g., her essay work, her dream) is NOT creditable.
- Temporary emotional or situational states do NOT count as character traits. Descriptions such as "
↳ tired," "stressed," "discouraged," "weighed down," "depressed," or "sad" should receive Score
↳ 0 even if story details are provided.
- However, "worried for her family" (or "concerned for her family") IS creditable when the response
↳ ties the worry to Rose's family rather than presenting it as a passing mood. Treat this as an
↳ enduring caring/protective disposition.
- When a student says Rose "is responsible" or "feels responsible for" her family, treat this as
↳ explicitly naming the trait "responsible." Do NOT downgrade to Score 0 merely because the
↳ same sentence also references Rose's burden, weight, or lack of time; the trait word itself
↳ is present and creditable.
- Traits that contradict the passage or are not textually supported (e.g., "shy," "selfish," "
↳ procrastinator," "leader," "brave," "honest," or "dreamer") are not creditable. Note
↳ specifically: "honest" is NOT defensible from these conversations because Rose deliberately
↳ chooses NOT to tell Aunt Kolab the truth about feeling weighed down.
- Traits such as "optimistic," "hopeful," "positive," "reassuring," "assuring," or "strong for her
↳ family" ARE creditable when grounded in Rose's statements to Anna about the family's future (
↳ e.g., that Paul will finish college and help pay for the sisters' schooling, or that things
↳ may get easier). Do not reject these traits merely because Rose is also shown to be burdened
↳ elsewhere in the passage.

****What qualifies as an adequate supporting detail (required for Score 2):****

- The detail should be drawn from the scenes containing Rose's conversations with Anna or Aunt Kolab.
↳ This includes: (a) what Rose or the other character says in the dialogue, (b) Rose's actions
↳ and behavior during or immediately surrounding those exchanges (e.g., checking on Anna,
↳ choosing not to burden her aunt with the truth, gently explaining why Papa had to leave,
↳ helping Aunt Kolab with supper), and (c) the circumstances Rose herself references in those
↳ conversations.

- Guidance on Rose's daily routine as supporting detail: A general recounting of Rose's schedule (
 - ↳ school + restaurant shift + helping Aunt Kolab with supper + homework) by itself is typically
 - ↳ NOT sufficient for the second point. Such recitations, even when accurate, function as
 - ↳ background context rather than support that clearly ties the trait to the conversation scenes.
 - ↳ To earn Score 2, the response should go beyond a generic schedule and include at least one
 - ↳ of the following: (i) a reference to or paraphrase of something Rose says in the dialogue
 - ↳ with Anna or Aunt Kolab, (ii) a reference to a specific action Rose takes within those
 - ↳ conversation scenes (e.g., checking on Anna, choosing not to answer her aunt truthfully,
 - ↳ nodding as her aunt offers to help), or (iii) an explicit link between a routine element and
 - ↳ Rose's conversation (e.g., naming that she comes home to help Aunt Kolab prepare supper as
 - ↳ part of what happens around the exchange, tied clearly to the trait). Score 1 is appropriate
 - ↳ when the trait is named but the only support offered is a general schedule recitation with no
 - ↳ such tie.
- Brief references, accurate paraphrases, and partial quotes are acceptable if they clearly connect
 - ↳ to the identified trait; a direct verbatim quotation is not required. A single short quote or
 - ↳ one-sentence reference from the conversation (e.g., quoting Rose asking Anna "You feel okay
 - ↳ ?" as evidence of caring) IS sufficient for the second point. Do NOT withhold Score 2 on
 - ↳ grounds that the supporting detail is "too sparse," "minimal," "brief," or "underdeveloped"
 - ↳ when the connection between the detail and the named trait is reasonably clear.
- Minor factual slips or slightly imprecise interpretations do not disqualify an otherwise adequate,
 - ↳ textually grounded supporting detail. Do not withhold the second point simply because a
 - ↳ stronger detail exists elsewhere in the passage.
- Details drawn only from Rose's dream or her essay-writing struggles (with no tie to the
 - ↳ conversations or the surrounding scenes) do NOT qualify.
- Rose's decision NOT to answer her aunt truthfully (because the truth would hurt her aunt) IS an
 - ↳ adequate detail for traits such as caring, considerate, respectful, thoughtful, protective,
 - ↳ or selfless. Do not reject this detail merely because the passage frames Rose's feelings as "
 - ↳ weighed down"—the student is citing Rose's protective silence, not claiming "weighed down" as
 - ↳ the trait.
- The student must EXPLICITLY name the trait in the response as a word or short phrase (e.g., "caring
 - ↳ ," "hardworking," "considerate"). Do NOT infer a trait from supporting details or behavior
 - ↳ descriptions alone. For example, a response that says "she doesn't like to hurt people's
 - ↳ feelings" or "she doesn't give up" without explicitly naming a trait like "considerate" or "
 - ↳ determined" should receive Score 0, even if the behavior described would fit a creditable
 - ↳ trait.