AI as a Calibration Anchor: Improving Grading Consistency in Large Introductory Programming Courses

Kareem Ibrahim
University of Toronto
Department of Electrical and Computer Engineering
Toronto, Ontario, Canada
kareem.ibrahim@mail.utoronto.ca

Salma Emara
University of Toronto
Department of Electrical and Computer Engineering
Toronto, Ontario, Canada
salma@ece.utoronto.ca

Abstract

Large introductory programming courses often rely on multiple teaching assistants (TAs) to grade open-ended submissions, where partial-credit decisions can vary even under shared rubrics. This experience report describes a human-in-the-loop AI-assisted grading workflow designed to make grading expectations more explicit to ensure consistency between TAs while preserving TA authority over final marks and feedback. The workflow combines instructor-authored rubrics with large language model-generated expansions that surface alternative correct solutions, common errors, and partial-credit cases. These are reviewed by instructors and refined through early TA disagreements before AI-generated marks and feedback are used as decision support that TAs can accept, revise, or reject.

We deployed this workflow across two large introductory programming courses, analyzing final TA grades, AI suggestions, comparable pre-AI grading data where grading relied solely on instructor-generated rubrics, grading-time records, number of accepted remark requests, and anonymous survey responses from TAs. Across comparable programming questions from prior offerings, mean inter-TA grading variability decreased from 4.97% to 2.16% (56.5% reduction). Across 7,899 programming AI-TA grade pairs, AI suggestions differed from final TA grades by 2.43% of the maximum question score on average. We also observe higher grading speeds in some cases, and lower change in grade for accepted remark requests. Survey responses highlighted practical benefits: 94% of TAs reported that AI helped catch grading oversights, and 82% reported increased confidence in final grades and notes. Overall, this report illustrates a practical model for AI-assisted grading at scale: improving consistency through shared calibration without replacing human judgment.

CCS Concepts

• **Applied computing** → **Computer-assisted instruction**; • **Social and professional topics** → **Student assessment**; • **General and reference** → *Empirical studies.*

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
SIGCSE Virtual 2026, Virtual Event, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2506-7/2026/11
https://doi.org/10.1145/3795867.3830965

Keywords

AI-assisted grading, grading consistency, human-in-the-loop systems

ACM Reference Format:
Kareem Ibrahim and Salma Emara. 2026. AI as a Calibration Anchor: Improving Grading Consistency in Large Introductory Programming Courses. In *Proceedings of the 2nd ACM Virtual Global Computing Education Conference V.1 (SIGCSE Virtual 2026), November 12–15, 2026, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3795867.3830965

1 Introduction

In large-enrollment courses, grading responsibilities for open-ended questions are often distributed across multiple teaching assistants (TAs). Many of these questions are open-ended and may allow for multiple valid approaches, partial solutions, and ambiguous intermediate reasoning. As a result, maintaining consistent grading across graders becomes challenging in practice, raising concerns about fairness and reliability. This challenge is especially relevant in introductory computer science courses, where programming exams often assess not only final answers but also students' reasoning, partial implementations, and understanding of core concepts.

Traditionally, such exams are graded using instructor-generated marking schemes and model answers that serve as a shared reference for TAs. However, prior work shows that human grading lacks a single gold standard, and that shared rubrics alone do not ensure consistent grading across evaluators [9]. TA experience, grading philosophy, and unanticipated edge cases can produce variation, while instructor arbitration does not scale well to large courses.

Alternative approaches, such as assigning multiple graders to the same submission, can improve consistency but are limited by grading resources and do not scale well with enrollment [7]. More recently, large language models have been proposed for automated grading; however, their outputs can be inconsistent or misaligned with course expectations without human oversight [2]. Taken together, these limitations highlight the **need for scalable mechanisms that align grading decisions in large courses.**

In this work, we introduce a human-in-the-loop AI-assisted grading workflow designed to improve consistency across TAs by making grading expectations more explicit. The approach uses a large language model to expand instructor-authored rubrics by surfacing alternative correct solutions, common errors, and partial-credit cases, creating a shared and more comprehensive reference for grading. This reference is iteratively refined using early TA grading cases and then used to generate suggested marks and feedback for individual submissions. TAs retain full authority over final decisions, using these suggestions as decision support rather than

automated grading. In this way, the system functions as a calibration mechanism that helps align grading behavior across TAs while preserving human judgment.

We deployed this workflow in two large introductory programming courses (CS1 and CS2) with enrollments of 400–440 students. These courses include programming exams with a mix of automatically graded questions and open-ended questions graded by TAs. We compare AI-assisted grading in CS2 (Fall 2025) and CS1 (Winter 2026) against comparable offerings from the previous academic year, where grading relied on instructor-generated rubrics and model answers without AI support.

Comparing comparable programming questions across offerings, we observe a 56.5% decrease in mean inter-TA variability. AI suggestions differed from final TA grades by 2.43% of the maximum question score on average, and surveyed TAs reported that AI helped identify grading oversights and increased confidence in grading decisions. These results suggest that AI can support grading consistency as a shared calibration aid rather than an automated replacement for TAs.

2 Related Work

Ensuring consistency in grading open-ended programming questions remains a significant challenge, particularly in large courses where multiple TAs grade responses to the same question. Prior work has explored multiple strategies to address inconsistency through three main approaches: rubric design, automated grading, and structured human grading workflows. Rubric-based approaches standardize evaluation by defining explicit criteria and guiding graders through structured decision-making; tools such as Rubyric [3] and ALOHA [1] enforce rubric use, while analytic and task-specific rubrics can improve reliability when paired with examples, training, or clearer performance levels [4, 5, 7]. However, rubrics primarily standardize criteria, and interpretation differences can persist when individual graders apply those criteria in practice.

Automated and semi-automated grading tools complement rubrics by improving scalability, but they often evaluate correctness through unit testing, code similarity, or rubric-level predictions. They are less directly suited to open-ended partial-credit decisions involving incomplete reasoning, design choices, or conceptual understanding [6, 10]. Recent survey work similarly shows that instructors want autograders that are customizable, integrated, and pedagogically extensible rather than limited to test-driven pass/fail judgments [14]. Thus, although automation improves scalability and consistency, it remains limited for open-ended grading tasks that require nuanced partial-credit reasoning.

Studies of autograder impact on learning show both benefit and risk: immediate feedback can support learning, confidence, and testing-skill development, but grading mistakes can harm learning when students fail to detect false positives or false negatives [8, 11]. These findings reinforce the need for human oversight when integrating automated or AI-assisted grading into educational contexts.

Another line of work focuses on process-oriented interventions, including double grading, calibration, and structured workflows. Double grading, adjudication thresholds, reliability metrics such as ICC, grader training, calibration discussions, and rubric refinement can improve agreement by aligning graders before or after grading [7]. However, empirical evidence shows that even with these interventions, graders exhibit very low agreement and substantial individual inconsistency, suggesting that achieving a reliable “gold standard" in human grading is inherently difficult [9]. These approaches also require substantial additional human effort, making them difficult to scale in large-enrollment settings.

Recent large language model (LLM)-assisted grading workflows provide suggested marks or rubric ratings for human review, improving grading efficiency while maintaining acceptable accuracy or agreement with manual grading [2, 12]. However, this line of work primarily evaluates efficiency, accuracy, regrade rates, or AI-human agreement, rather than how AI assistance affects consistency across multiple graders working in shared grading contexts.

Across these lines of work, prior approaches either standardize criteria, automate evaluation, or improve agreement through additional human effort, but they do not directly address how to align multiple graders’ interpretations during grading [13]. This gap is critical in large courses, where consistency depends not only on shared rubrics, but on shared interpretation.

3 Methodology

3.1 Course and Question Types Context

We report deployments in two large introductory programming courses, CS 1 and CS 2, offered during 2025–2026, each enrolling 400—450 students. We compare these offerings with the same courses in 2024—2025, before the AI-assisted workflow was used.

The exam structure in all course offerings was computer-based and included two main parts: (i) true/false and multiple choice questions testing conceptual understanding, which were graded automatically, and (ii) explanation, debugging and programming questions, which were graded by TAs and are the focus of this paper. Depending on grading needs and TA availability, between 1 and 5 TAs were assigned to grade each question. Students were permitted to submit a regrade request within one week after grades were released in all course offerings.

The TA-graded questions required more than checking whether student code passed predefined test cases. Explanation and debugging questions asked students to explain or find/fix bugs in provided code, and programming questions asked students to create code that functioned as desired. Student solutions reflected different levels of understanding and reasoning, and hence could be partially correct, incomplete, or unconventional. As a result, graders had to interpret student intent, award partial credit, and write comments that were fair and consistent even when answers differed substantially in structure and clarity.

3.2 Baseline Grading Workflow

In 2024–2025 academic year, we were solving this practical consistency problem through traditional methods. Instructors created a model answer and rubric for each question that was shared among TAs grading the same question. While each TA received the same rubric, often for programming questions, TAs find solutions that follow a different approach. Instructors ask TAs to extend the rubric when new approaches are observed to have other TAs follow them consistently. Usually a TA would independently make a judgement

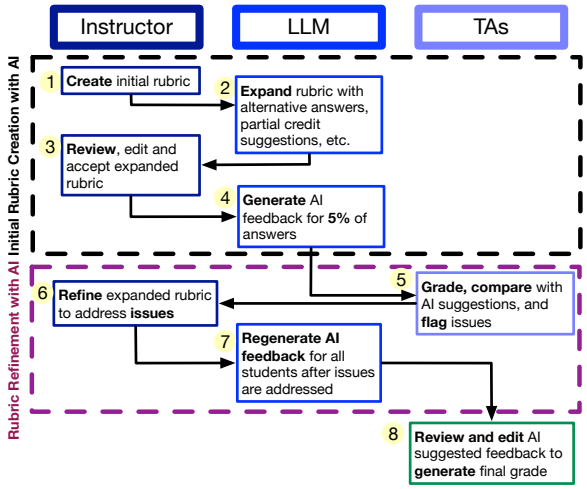

Figure 1: Human-in-the-loop AI-assisted grading workflow, where the instructor and TAs help refine the expanded rubric used for grading with the LLM.

on how to follow the rubric, and ideally they discuss among other TAs if they see a different approach before making a judgement. Discussions may not take place if TAs are grading asynchronously.

Despite shared rubric language, edge cases, rare mistakes, and unusual but valid approaches were often resolved locally by individual TAs. This made it difficult to ensure that similar solutions received similar treatment, especially once grading was underway and instructor arbitration became hard to scale.

3.3 AI-Assisted Grading Workflow

Because our exam questions required partial-credit reasoning rather than simply checking whether code passed predefined tests, we designed the 2025–2026 workflow to support TA decision-making rather than automate grading.

Fig. 1 summarizes the workflow. For each question, the instructor first created the standard grading materials: a model answer and an initial rubric. These materials were then used to prompt the large language model, *i.e.*, `gpt-5.4-thinking`, to expand the grading guidance by surfacing:

- alternative correct answers,
- likely student errors, and
- plausible partial-credit cases that might not be captured in the initial rubric alone.

The instructor reviews this expanded guidance, edits it if the LLM is overly lenient or strict according to their expectations, and approves the version to be used operationally. A lower-cost LLM, *i.e.*, `gemini-3.1-pro`, was then prompted to generate rubric-aligned feedback and a suggested grade for a 5% sample of student responses.

Multiple TAs grading the question start by grading without looking at LLM-generated feedback and grade, and then they compare their judgement with the AI exposing disagreement cases, missing edge cases, or AI behavior that was too strict, too lenient, or insufficiently focused. TAs and instructors then consolidated these observations to refine the shared guidance. After refinement, AI-generated grade suggestions and notes were regenerated and made available for TA review on the remaining student responses.

The key feature of this workflow was the **refinement step**. The goal was not to ask the model for one-shot judgments and then treat those outputs as stable. Instead, instructors and TAs used early grading experience to iteratively improve the shared reference that shaped future AI suggestions and TA decisions. In practice, AI served as a coordination mechanism, helping TAs apply the rubric more consistently to partial-credit cases while retaining full authority to accept, revise, or reject AI suggestions.

3.4 Study Design and Data Collection

We report on two real deployments of this AI-assisted workflow in CS 1 and CS 2 as an observational experience report focused on how the workflow was developed, used, and refined in authentic course settings. Our evidence comes from regular course operation, including grading artifacts, instructor decisions, TA interactions, AI-generated guidance, regrade outcomes, timing logs, and post-grading reflections. Accordingly, our analysis emphasizes implementation details, operational outcomes, practical value, and limitations that may inform adoption in similar large programming courses. This study was reviewed and approved by the institutional research ethics review board prior to data analysis and reporting.

The quantitative component combined de-identified course-operation data, including final TA-assigned grades, AI-suggested grades, TA allocation data, comparable pre-AI offerings from 2024–2025, limited grading-time logs, and accepted remark-request outcomes. The primary quantitative evidence comes from multi-TA programming-question slices, because these cases most directly address the consistency problem motivating the workflow; single-TA and short-answer cases are used more cautiously as supporting evidence for TA-AI alignment. The qualitative evidence drew on an anonymous voluntary TA survey and short structured instructor interviews, which were used to contextualize how the workflow supported consistency, where it remained fragile, and why refinement was necessary before AI suggestions became operationally useful.

We interpret the quantitative evidence cautiously: (i) inter-TA consistency is treated as between-TA dispersion on the same question under shared guidance, (ii) TA-AI comparisons are interpreted as evidence that AI suggestions provided a useful shared reference for grading, not as evidence of grading accuracy against a gold standard, and (iii) comparisons to pre-AI offerings are interpreted as suggestive rather than causal because the assessments are comparable rather than identical.

4 Evaluations

We evaluate AI-assisted grading along four main dimensions: (1) inter-TA consistency, (2) AI-TA grading agreement, (3) grading efficiency, and (4) outcomes of accepted remark requests.

4.1 Impact of AI on Inter-TA Consistency

We study the impact of using AI-assisted grading on consistency among multiple TAs grading the same question. Given that questions graded by multiple TAs were always programming questions and not short answer, we focus on programming questions. We

sampled 28 programming questions: 14 graded with AI support and 14 graded without AI support. In each group, we had 7 from CS 1 and 7 from CS 2. For each question, there were 2 to 7 TAs grading.

To quantify grading consistency across TAs, we define a measure of *Inter-TA Severity Variability* (σ_q^{norm}) for each question as the weighted standard deviation of TA-level average grades, normalized by the maximum possible score for that question:

$$\sigma_q^{norm} = \frac{1}{M_q}\sqrt{\frac{\sum_{i=1}^{k} n_i\,(g_i - \bar{g})^2}{\sum_{i=1}^{k} n_i}} \tag{1}$$

where g_i denotes the average grade assigned by TA i, n_i is the number of submissions graded by TA i, k is the total number of TAs who graded the question, and M_q is the maximum possible score for question q. The weighted mean grade $\bar{g}$ is defined as:

$$\bar{g} = \frac{\sum_{i=1}^{k} n_i g_i}{\sum_{i=1}^{k} n_i}. \tag{2}$$

This formulation accounts for uneven grading workloads by weighting each TA's contribution according to the number of submissions they graded. It also accounts for different question maximum scores by scaling the between-TA standard deviation by M_q, allowing variability to be compared across questions with different maximum scores. This metric captures systematic differences in grading severity (i.e., relative leniency or strictness) across TAs for the same question. A higher σ_q^{norm} indicates greater variability in grading behavior relative to the question's maximum score, while a lower value indicates stronger agreement among TAs.

Fig. 2 shows the distribution of between-TA severity variability (σ_q^{norm}) with and without AI support. Lower values indicate stronger agreement among TAs.

Across all questions, AI-assisted grading substantially reduces inter-TA variability. The mean σ_q^{norm} decreases from 4.97% without AI to 2.16% with AI, representing a 56.5% reduction. A similar trend is observed in the median (5.02% without AI vs. 2.05% with AI). A Mann–Whitney U test confirms that this reduction is statistically significant ($U = 33.0$, $p = 0.003$), supporting the interpretation that AI support is associated with more consistent grading across TAs.

When disaggregated by course, the effect is particularly pronounced in CS 2. The mean σ_q^{norm} decreases from 5.55% without AI to 2.05% with AI, with medians of 5.84% and 2.57%, respectively. This difference is statistically significant ($U = 7.0$, $p = 0.0298$), suggesting that AI substantially improves grading consistency in this course.

In CS 1, we also observe a reduction in inter-TA variability, with the mean σ_q^{norm} decreasing from 4.39% without AI to 2.27% with AI, and medians decreasing from 3.38% to 1.54%. However, this difference is not statistically significant ($U = 10.0$, $p = 0.0736$). This suggests that AI may improve consistency in CS 1, but the effect was smaller than in CS 2, possibly because CS 1 programming questions more explicitly guided students through solution steps, leaving less baseline variability for AI to reduce.

Overall, these results indicate that AI-assisted grading consistently reduces variability in TA grading severity. These findings suggest that AI primarily acts as a calibration mechanism, aligning TA grading behavior and reducing systematic differences in grading severity. The strongest and most statistically robust effects observed in CS 2 where the questions were more open-ended and allowed more solution approaches.

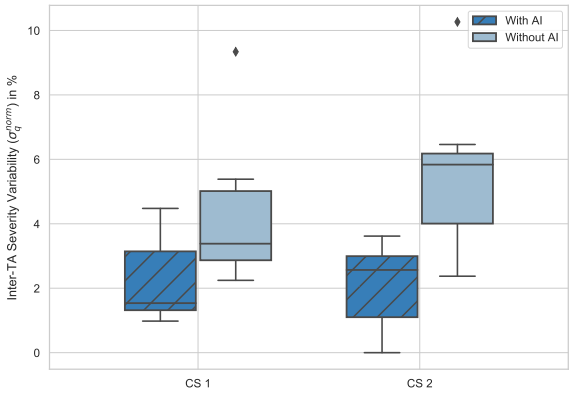

Figure 2: Between-TA severity variability (σ_q^{norm}) with and without AI support across courses. Lower values indicate greater consistency among TAs.

4.2 AI–TA Agreement

To better understand how AI suggestions related to TA decisions, we evaluate AI–TA grade agreement. Across 7,899 AI–TA grade pairs for programming questions, exact agreement was 82.8%. To dive deeper into when AI and TA grades do not agree, we calculate the mean absolute error (MAE) for each question, which is the absolute difference between the AI suggested grade (g_i^{AI}) and TA grade (g_i^{TA}) for submission i averaged over all k submissions for a question. Because questions varied in maximum score, in Eqn. 3, we normalized the MAE by dividing it by the maximum possible score for that question (M_q) and we report MAE in %.

$$\text{MAE} = \frac{1}{k}\sum_{i=1}^{k}\left|\frac{g_i^{AI} - g_i^{TA}}{M_q}\right| \tag{3}$$

When MAE is 0%, it means perfect agreement, and the higher the number the less aligned the TA is with the AI. On average, we observe the normalized MAE is 2.43%, which means the average difference between the AI grade and TA grade is 2.43% of the maximum score. This difference is small, especially given that marks were assigned in one-point increments rather than fractional increments.

Figure 3 summarizes AI–TA alignment across courses and question types. We evaluate alignment on the same programming questions used in the inter-TA consistency analysis, and include additional short answer questions to broaden coverage.

At the course level, AI demonstrates substantially stronger alignment with TAs in CS 1 than in CS 2. The average normalized MAE is 1.63% in CS 1 compared to 3.23% in CS 2 (both 10 questions considered). This difference is statistically significant (Mann–Whitney $U = 17$, $p = 0.01$) suggesting a consistent trend toward better alignment in CS 1.

When disaggregated by question type, AI achieves lower MAE on short answer questions than on programming questions in both courses. In CS 1, the average normalized MAE is 1.05% for short

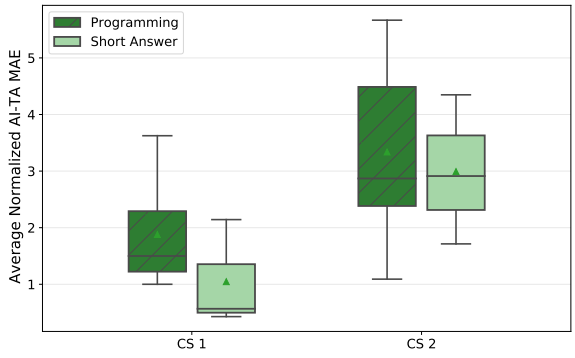

Figure 3: AI-TA agreement measured by normalized mean absolute error (MAE) between AI and TA grades across course and question type. Lower MAE indicates stronger agreement.

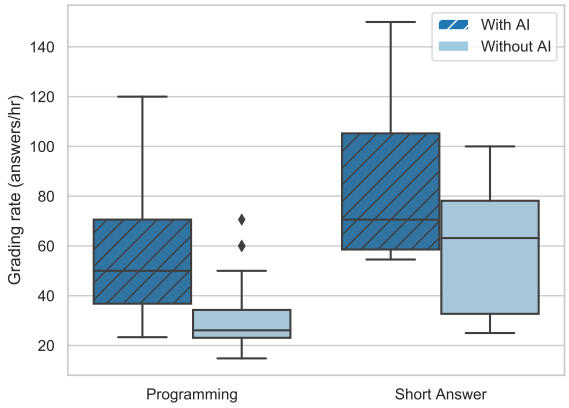

Figure 4: Grading rates (answers/hour) for programming and short-answer questions, comparing AI-assisted and non-AI conditions for CS 1.

answer questions compared to 1.88% for programming questions. A similar pattern is observed in CS 2, where short answer questions yield an average normalized MAE of 2.99% compared to 3.34% for programming questions. This suggests that TA grades agree more closely with AI suggested grades on questions with more constrained responses.

Overall, these results indicate that AI achieves consistent agreement with TA grading, with stronger agreement observed in CS 1 (than CS 2) and in short answer questions (than programming).

4.3 Impact of AI on Grading Efficiency

We sampled 33 questions for which grading time was tracked. There were multiple TA graders per question, and we calculated the rate of grading per TA per question.

Overall, the average grading rate increased from 46.9 answers/hour without AI to 58.4 answers/hour with AI. However, given there are differences in question types and their level of difficulty, we decided to dive deeper into each course and different question types.

In Table 1, we present grading rates (mean, median, IQR) across question types and AI conditions for both courses. For CS 1, the median grading rate for programming questions was 89% higher with AI (50.0 answers/hour) than without AI (26.1 answers/hour), and a Mann–Whitney U test showed that this difference was significant ($U = 61.5$, $p = .002$). For short-answer questions, median grading rates were higher with AI (70.59 answers/hour) than without AI (63.16 answers/hour), but this difference was not statistically significant ($U = 17$, $p = .33$). This suggests that AI did not uniformly improve grading productivity, especially for short-answer questions. One possible explanation is that, for questions already quick to grade, reading and verifying AI-generated feedback may offset time savings. We summarize the CS 1 results in Fig. 4.

For CS 2, the median for the grading rate of short answer and programming questions was slightly different, but the difference was not statistically significant (for programming $U = 7, p = .43$, short answer $U = 9, p = .83$). We do not draw a strong conclusion for CS 2 because grading time was tracked for fewer questions.

Table 1: Grading rates (answers/hour) across question types and AI conditions for CS1 and CS2.

CS1					
Question Type	AI	Mean	Median	IQR	N (TAs)
Programming	Yes	58.70	50.00	33.79	15
Programming	No	32.64	26.09	11.21	21
Short Answer	Yes	86.11	70.59	46.67	7
Short Answer	No	58.56	63.16	45.44	7
CS2					
Question Type	AI	Mean	Median	IQR	N (TAs)
Programming	Yes	25.43	22.29	14.55	7
Programming	No	27.38	28.57	1.79	3
Short Answer	Yes	74.44	74.44	11.28	2
Short Answer	No	73.59	72.24	53.85	8

4.4 Impact on Remark Requests Results

In CS 1, the number of accepted remark-request questions decreased by 50% when the AI-assisted workflow was used, from 26 to 13. However, for CS 2, the number of accepted remark request questions increased slightly from 27 to 32. Since accepted remark counts depend partly on student submission behavior, we also examine the average grade change among accepted requests as an indicator of how much initial grades shifted after review. In CS 1, the average change in grade across all accepted requests dropped from 2.4 to 1.2 (50%). In CS 2, the drop was from 2.4 to 1.5 ($\approx$ 38% drop).

4.5 TA Perceptions of AI-Assisted Grading

We complement our quantitative analysis with anonymous survey responses from 17 TAs across CS 1 and CS 2. While the sample size is limited, the responses provide qualitative insight into how AI feedback was used in practice and help contextualize observed trends in consistency, alignment, and grading efficiency.

With respect to inter-TA consistency, respondents strongly indicated that AI feedback helped standardize grading decisions. In particular, 16 of 17 TAs (94.1%) agreed that AI helped them catch grading oversights, such as missed rubric items or partial-credit opportunities. Additionally, 82.4% reported increased confidence in their assigned grades. These findings support the observed reduction in between-TA variability, suggesting that AI may reduce inconsistency by highlighting rubric-relevant details and providing a shared reference point for grading decisions.

When asked where AI feedback was most useful, the top three scenarios selected by more than 50% of TAs were phrasing feedback clearly (70.6%), evaluating borderline partial-credit answers (58.8%), and assessing unusual solution approaches (52.9%). When asked where AI feedback struggled, the most common responses were that it generally performed well (64.7%), struggled with handling alternative correct methods (41.2%), and with assigning partial credit fairly (35.3%). These limitations align with the AI–TA disagreements observed for programming questions, where solutions are more open-ended and rubric interpretation is less constrained.

Perceptions of grading efficiency were generally positive but nuanced. A large majority of respondents (88.2%) reported that grading felt faster with AI, and an equal proportion indicated that AI saved time when writing student-facing feedback. However, nearly half of respondents (47.1%) noted that AI introduced additional time due to the need to verify generated feedback. This tradeoff helps explain the mixed efficiency results across courses: while AI reduces effort in feedback generation, the need for human oversight may offset time savings in more complex grading contexts.

Importantly, responses to the open-ended general-feedback question consistently described AI as a decision-support tool rather than an authoritative grader. When disagreements arose, most TAs relied on their own judgment or consulted peers. One respondent expressed concern that some graders might defer too readily to AI. This pattern reinforces the interpretation of AI as a calibration aid that supports, rather than replaces, human grading. Overall, the survey results align with our quantitative findings and suggest that AI is most effective when used as a human-in-the-loop tool to improve consistency and support grading decisions in structured contexts.

4.6 Instructor's Perspective and Limitations

From the instructor's perspective, the AI-assisted workflow was valuable not only because it produced grade suggestions, but because it made grading assumptions more explicit. Reviewing AI-expanded rubrics helped instructors identify alternative correct approaches, common misconceptions, and partial-credit cases that were missing or under-specified in the original rubric. This allowed instructors to refine grading guidance before ambiguities became inconsistent TA-level decisions.

The AI-generated feedback was also useful as a student-facing communication tool. When reviewed by TAs, it helped translate rubric deductions into clearer explanations of what was missing or misunderstood. This was especially important because, in prior offerings, time constraints meant that some students received little or no meaningful feedback beyond the assigned mark. Even imperfect reviewed feedback can be pedagogically valuable, because it gives students a starting point for understanding their mistakes rather than leaving them to infer the reason for lost marks on their own.

Overall, from the instructor's perspective, AI-supported feedback helped make grading more transparent, more consistent, and more useful for student learning while preserving human judgment over final grades and comments.

As an experience report based on authentic course deployments, our findings should be interpreted in light of the course contexts studied. The comparisons across years involved comparable but non-identical assessments, and the TA cohorts differed across offerings. While these comparisons reflect realistic instructional conditions, differences in question design, rubric detail, student responses, or grader experience may have contributed to the observed outcomes. The workflow also combines several mutually reinforcing components, including AI-expanded and refined rubrics, instructor and TA reviews. Therefore, the results reflect the overall human-in-the-loop workflow rather than any single component in isolation.

We interpret inter-TA consistency as evidence of stronger grading alignment rather than as a direct measure of grading accuracy. AI–TA agreement similarly indicates that AI suggestions provided a practical common reference for graders, while still requiring instructor and TA oversight. Finally, timing logs, remark-request outcomes, and TA survey responses provide useful operational evidence, but are naturally shaped by course practices, student behavior, and voluntary participation. These limitations contextualize the findings while preserving the main takeaway: the workflow helped make grading expectations more explicit and supported more consistent human grading in realistic large-course settings.

5 Conclusion

In these two deployments, AI was most useful as a shared calibration aid rather than as an autonomous grader. The critical design choice in our workflow was the refinement step: TAs first graded a small sample independently, identified AI-TA disagreement cases, and then used these observations to refine the shared rubric before AI feedback was regenerated for all students. This step transformed AI from a one-shot grader into an anchor around which instructors and TAs could make grading assumptions explicit and converge on a shared interpretation of the rubric. In settings where open-ended programming answers vary widely, this matters because consistency depends not only on whether TAs receive the same rubric, but on whether they apply it similarly across partial-credit cases, unusual solutions, and ambiguous student reasoning.

Across two large introductory programming courses, this anchor-based workflow was associated with substantially more stable grading. Mean normalized inter-TA severity variability decreased from 4.97% without AI to 2.16% with AI, a 56.5% reduction, and AI–TA grade pairs showed 82.8% exact agreement with only 2.43% normalized MAE. These gains were accompanied by practical benefits: 94.1% of surveyed TAs reported AI helped them catch grading oversights, 82.4% reported increased confidence in their grades, and accepted remark requests led to smaller grade changes than in prior offerings. Overall, our experience suggests that AI-assisted grading is most useful as a calibration and coordination mechanism for making grading expectations more explicit in large courses.

References

[1] Tuukka Ahoniemi and Ville Karavirta. 2009. Analyzing The Use of a Rubric-Based Grading Tool. In *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education* (Paris, France) *(ITiCSE '09)*. Association for Computing Machinery, New York, NY, USA, 333–337. doi:10.1145/1562877.1562977

[2] Jared Apillanes, Jason Weber, Sergio Gago-Masague, Jennifer Wong-Ma, and Thomas Y. Yeh. 2026. Fighting Fire with Fire: LLM-Assisted Grading of Handwritten CS Assessments. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education (SIGCSE)*. Association for Computing Machinery, New York, NY, USA, 66–72.

[3] Tapio Auvinen. 2011. Rubyric. In *Proceedings of the 11th Koli Calling International Conference on Computing Education Research* (Koli, Finland). Association for Computing Machinery, New York, NY, USA, 102–106.

[4] Jingwen Bai, Wei Soon Cheong, Philippe Muller, and Brian Y Lim. 2026. iRULER: Intelligible Rubric-Based User-Defined LLM Evaluation for Revision. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA. doi:10.1145/3772318.3790539

[5] Alan Chong and Lisa Romkey. 2017. Testing Inter-Rater Reliability in Rubrics for Large Scale Undergraduate Independent Projects. *Proceedings of the Canadian Engineering Education Association* (01 2017). doi:10.24908/pceea.v0i0.6465

[6] Jayant Havare, Varsha Apte, Kaushikraj Maharajan, Nithin Chandra Gupta Samudrala, Ganesh Ramakrishnan, Srikanth Tamilselvam, and Sainath Vavilapalli. 2025. AI-Based Automated Grading of Source Code of Introductory Programming Assignments. In *2025 IEEE/ACM 33rd International Conference on Program Comprehension (ICPC)* (Ottawa, ON, Canada). 171–181.

[7] Wei Jin, Xin Xu, Hyesung Park, Evelyn Brannock, and Tacksoo Im. 2026. Improving the Reliability of Grading Written-Response Coding Questions in a Large CS1 Course. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education (SIGCSE)*. Association for Computing Machinery, New York, NY, USA, 547–553.

[8] Tiffany Wenting Li, Silas Hsu, Max Fowler, Zhilin Zhang, Craig Zilles, and Karrie Karahalios. 2023. Am I Wrong, or Is the Autograder Wrong? Effects of AI Grading Mistakes on Learning. In *Proceedings of the 2023 ACM Conference on International Computing Education Research (ICER)* (Chicago, IL, USA) *(ICER '23)*. Association for Computing Machinery, New York, NY, USA, 159–176.

[9] Marcus Messer, Neil Brown, Michael Kölling, and Miaojing Shi. 2025. How Consistent Are Humans When Grading Programming Assignments? *ACM Transactions on Computing Education (TOCE)* 25, 4 (2025), 1–37.

[10] Marcus Messer, Neil C. C. Brown, Michael Kölling, and Miaojing Shi. 2024. Automated Grading and Feedback Tools for Programming Education: A Systematic Review. *ACM Transactions on Computing Education (TOCE)* 24, 1 (2024).

[11] Joydeep Mitra. 2023. Studying the Impact of Auto-Graders Giving Immediate Feedback in Programming Assignments. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education (SIGCSE 2023)* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 388–394.

[12] Goda Nagakalyani, Saurav Chaudhary, Varsha Apte, Ganesh Ramakrishnan, and Srikanth Tamilselvam. 2025. Design and Evaluation of an AI-Assisted Grading Tool for Introductory Programming Assignments: An Experience Report. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education (SIGCSE)* (Pittsburgh, PA, USA). Association for Computing Machinery, New York, NY, USA, 805–811.

[13] Kai Presler-Marshall, Sarah Heckman, and Kathryn T. Stolee. 2023. Improving Grading Outcomes in Software Engineering Projects Through Automated Contributions Summaries. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. 259–270. doi:10.1109/ICSE-SEET58685.2023.00030

[14] Barrett Ruth and John R. Hott. 2025. Auto-grading in Computing Education: Perceptions and Use. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education (SIGCSE)* (Pittsburgh, PA, USA). Association for Computing Machinery, New York, NY, USA, 1008–1014.