Feasibility Study of Augmenting Teaching Assistants with AI for CS1 Programming Feedback

Umair Z. Ahmed
umair@nus.edu.sg
National University of Singapore
Singapore

Shubham Sahai
shubham@nus.edu.sg
National University of Singapore
Singapore

Ben Leong
benleong@comp.nus.edu.sg
National University of Singapore
Singapore

Amey Karkare
karkare@cse.iitk.ac.in
Indian Institute of Technology Kanpur
Kanpur, India

Abstract

With the increasing adoption of Large Language Models (LLMs), there are proposals to replace human Teaching Assistants (TAs) with LLM-based AI agents for providing feedback to students. In this paper, we explore a new hybrid model where human TAs receive AI-generated feedback for CS1 programming exercises, which they can then review and modify as needed. We conducted a large-scale randomized intervention with 185 CS1 undergraduate students, comparing the efficacy of this hybrid approach against manual feedback and direct AI-generated feedback.

Our initial hypothesis predicted that AI-augmented feedback would improve TA efficiency and increase the accuracy of guidance to students. However, our findings revealed mixed results. Although students perceived improvements in feedback quality, the hybrid model did not consistently translate to better student performance. We also observed complacency among some TAs who over-relied on LLM generated feedback and failed to identify and correct inaccuracies. These results suggest that augmenting human tutors with AI may not always result in improved teaching outcomes, and further research is needed to ensure it is truly effective.

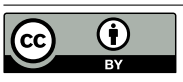
CCS Concepts

• **Social and professional topics** → **CS1**; • **Applied computing** → **Computer-assisted instruction**.

Keywords

CS1, Programming, Randomized Trial, TA, Hint, LLM, GPT

ACM Reference Format:
Umair Z. Ahmed, Shubham Sahai, Ben Leong, and Amey Karkare. 2025. Feasibility Study of Augmenting Teaching Assistants with AI for CS1 Programming Feedback. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE TS 2025), February 26-March 1, 2025, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3641554.3701972



SIGCSE TS 2025, February 26-March 1, 2025, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0531-1/25/02
https://doi.org/10.1145/3641554.3701972

1 Introduction

The introductory programming course (CS1) is one of the most popular course across universities [20]. A large number of students struggle to master difficult concepts [24] and without adequate formative feedback, these students could lose motivation or even drop out of their courses [15]. However, the ever increasing enrollments [21] have led to a worsening student-tutor ratio, making it hard to provide such feedback manually at scale [7].

Recent advances in Large Language Models (LLMs) have demonstrated the potential to alleviate this problem by aiding struggling CS1 students in improving their learning outcomes in a controlled setting [23]. However, modern LLM-based approaches are susceptible to inaccurate feedback or worse hallucinations, presenting a challenge for educational settings [19].

Instead of solely relying on human Teaching Assistants (TAs) to provide feedback to CS1 students, or replacing them with an AI agent, we investigate a new hybrid model where human TAs are provided with AI-generated feedback which they can verify and edit before sending it to students. Our hypothesis is that such an approach could potentially address the problems of hallucination by LLM-based AI tutors and yet allow human TAs to become more efficient in responding to student queries.

The use of LLMs to generate feedback is a relatively new approach. Unlike deterministic methods for feedback generation [2, 10], LLMs allows flexibility in style of feedback through variations in prompting. It is conceivable that the style of feedback would have an impact on student learning. Hence, in this paper, we investigate both the impact of providing AI assistance to TAs and how different feedback styles in LLM-generated feedback can affect the effectiveness of this new approach.

To this end, we conducted one of the first large-scale user studies with 185 undergraduate CS1 students to explore the impact of AI-generated feedback on their live programming performance at IIT Kanpur, a large public university in India. To the best of our knowledge, this is also the first time that the impact of AI-generated feedback is compared against both feedback from traditional human TAs and human-vetted AI-generated feedback. In the process, we also investigated whether providing AI assistance to TAs would improve their performance and whether human TAs can effectively mitigate the hallucination arising from using an LLM in the feedback generation.

The following are the key insights from our user study:

(1) While students tend to prefer more direct feedback instead of Socratic-style feedback (§4.4), there was no significant difference in performance between students receiving default-style and Socratic-style AI feedback (§4.2);
(2) Students helped by TAs who had no access to AI assistance were found to perform significantly better than those who received feedback from hybrid tutors with access to AI-generated feedback. When we delved into the data, we found that this was because human TAs tended to be more crisp and focused in their feedback, occasionally giving away answers directly to the students (§4.3);
(3) Feedback accuracy does not necessarily translate into perceived helpfulness by the students, and student perception is not a direct measure of learning outcomes when it comes to AI-generated feedback (§4.4);
(4) Students often cannot identify wrong feedback arising from hallucination from LLMs. Experts identified these issues, rating them to be lower in accuracy and helpfulness, even though the students were satisfied with the feedback (§4.4);
(5) The qualitative evaluation of feedback revealed that hybrid TAs with AI assistance provided more accurate and helpful feedback but did not completely eliminate hallucination or invalid feedback generated by AI. This suggests a need for better training and integration strategies for AI assistance in educational contexts (§4.5); and
(6) While we expected TAs to become more efficient when they had access to AI-generated feedback, we found that surprisingly, TAs provided with Socratic-style AI feedback would tend to take significantly longer than TAs without AI support in serving student queries (§4.5).

Our work presents a preliminary investigation on the impact of augmenting human TAs with AI feedback. While we believe that such a hybrid approach is promising in providing feedback at scale for CS1, it is clear that doing so does not automatically make the TAs more effective or efficient. Our user study artifacts, containing the CS1 programming assignment dataset and experimental results, have been made publicly available to aid further research[1].

2 Related Work

AI in education has garnered significant attention for its potential to enhance learning outcomes by providing automated formative feedback to struggling students. Intelligent Tutoring Systems (ITS) have been proposed for decades, aiming to deliver personalized learning experiences based on student interactions [5]. However, the impact of AI-generated feedback has produced mixed results. For instance, VanLehn found that AI tutors can be as effective as human tutors under certain contexts [22], whereas Kulik et al. reported varying effectiveness of AI feedback, depending on the implementation and context of the AI systems [11]. Ahmed et al. conducted a large scale user study and found that CS1 students with access to automated repairs and examples of repairs outperformed their human tutored peers in resolving compilation errors, attributing the advantages to logistical rather than conceptual improvements [3].

The recent advancements in Large Language Models (LLMs) and their ability to process code have made them capable of generating more accurate and context-aware feedback [6, 18]. This has led to their widespread adoption in CS1 setting [12, 16, 17, 23]. Wang et al. reported on a large scale user study where students with access to LLM generated error messages took fewer attempts in resolving their compilation error, compared to students relying on standard error messages [23]. Liffiton et al. evaluated CodeHelp, an AI based tool that provides immediate feedback to students' programming queries and found it to improve engagement [12]. Similarly, Zamfirescu et al. found that an LLM-based AI homework assistant helped students complete their homework more quickly, with a significant impact on students who spent more time on their homework earlier without assistance [26]. Denny et al. explored student interactions with AI Teaching Assistants (TAs) and found that students engaged with it consistently, especially near assignment deadlines and outside office hours, preferring hints and guidance over direct solutions [9].

While the integration of AI for direct student feedback has been widely explored, its use in augmenting Teaching Assistants (TAs) has received less attention. Preliminary findings suggest potential benefits. For example, Yi et al. found that automated repairs as hints improved the performance of CS1 TAs in their grading tasks [25]. Markel et.al evaluated GPTeach to train novice teachers by facilitating interaction with GPT-simulated students and found it helpful for training without pressuring students [13].

In contrast, our study is the first real-world implementation of its kind to explore the feasibility and impact of AI-augmented TAs in a graded CS1 course. Unlike previous studies that focused on either providing direct AI feedback to students, or restricting the scope to specific errors such as compilation errors, or using the generated fixes as solution-level hints, our work focuses on the broader application of AI in assisting real-time TA-student interactions, and provides insights into both the benefits and challenges of AI. This study lays the groundwork for future research to optimize AI integration in educational settings to complement and enhance the human educators.

3 Experiment Design

To investigate the impact of AI assistance for human Teaching Assistant (TA), we built an AI agent [19] powered by GPT-4T [14] that can generate personalized feedback for buggy solutions for a CS1 course. Given (i) the student's query, (ii) the incorrect student program, (iii) the problem description, and (iv) the testcase evaluation results, our AI agent can generate feedback for the individual lines of code that are deemed to be "wrong." Our AI agent, built on Large Language Models (LLMs), is prone to hallucination, though at a low rate of around 9% for our use case. One of our goals was to investigate how human TAs would fare when AI feedback was imperfect.

We were fortunate to obtain permission to conduct a user study for a *graded* programming lab with 185 students at IIT Kanpur, a large public university in India. Human TAs are typically assigned to provide assistance to the students during these labs. We obtained permission to replace some of these tutors with an AI agent and also to provide AI feedback to assist some of these TAs. Appropriate IRB approval was sought from the participating institution for our study. The fact that we were working in a practical environment

[1] https://github.com/ai-cet/sigcse2025-userStudy-artifacts

Table 1: Distribution of experimental groups and sample feedback for each group. For Q1, students received AI-generated feedback. For Q2, all students received feedback from human TAs, with some TAs augmented by AI-generated feedback.

Question	Group	#Students	#Requests	%Completion	Sample Feedback
Q1	AI (Default)	23	44	82.6%	`(Line-20) The function 'printPosDir' is not implemented. Loop through the array of 'struct posDir' and print the position and direction . . .`
	AI (Socratic)	23	49	78.2%	`(Line-12) What structure or data type could you use to keep track of the traveler's current position and direction?`
	None	126	0	76.2%	-
Q2	TA-AI (Default)	5	7	40.0%	`(Line-42) You need to create a new node and assign the data to this node, then adjust the top pointer accordingly.`
	TA-AI (Socratic)	11	15	36.4%	`(Line-54) What checks should you perform before attempting to pop an element from the stack?`
	TA-Manual	13	17	38.4%	`(Line-69) For 'e' you have given two cases, so its calling two functions`
	None	130	0	56.2%	-

instead of with a group of volunteers provides us with a more accurate picture of how well AI assistants would work in a real life setting. Unfortunately, this also imposes certain constraints on the experiment that we could conduct and we needed to design our interventions carefully and with creativity.

Methodology. We conducted a randomized intervention trial with 185 undergraduate students enrolled in a CS1 C programming course. To facilitate this trial, we replaced the usual practice of human TAs circulating the room to assist students with an online interface through which students could request help. To ensure fairness, all students received reasonably equivalent forms of feedback throughout the session. The lab consisted of two questions on pointers and linked lists, which students had to complete within a 3-hour session consisting of a simpler question (Q1) and a more complex one (Q2). For Q1, all feedback was generated exclusively by an AI assistant. For Q2, feedback was either provided directly by a human TA or generated by an LLM, which was then reviewed and vetted by a TA before being sent to the students.

The LLM-generated feedback was provided in two distinct styles: (a) Default style, where no specific feedback style was instructed in the prompt, and (b) Socratic style, which offered guidance designed to encourage critical thinking and help students arrive at solutions independently. Figure 1 illustrates the workflow of our experiment, and Table 1 shows the distribution of students across experimental groups for the two questions.

Students were divided into the following six experimental groups:

(1) **AI (Default)**: The students in this group were provided with default-style hints generated by the LLM. These hints are designed to provide direct guidance and suggestions to the students without revealing the solution.
(2) **AI (Socratic)**: The students in this group were provided with Socratic-style hints generated by the LLM. These hints are designed to prompt students to think critically and guide them towards finding the solution on their own.
(3) **TA-AI (Default)**: The students were provided with hints by TAs who were assisted by the default-style hints generated by the LLM.
(4) **TA-AI (Socratic)**: The TAs in this group were assisted with LLM-generated abstract Socratic-style hints.

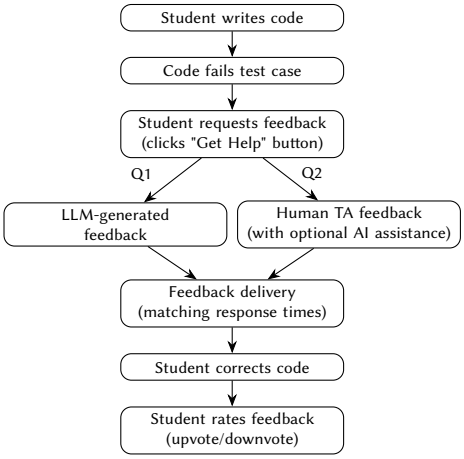
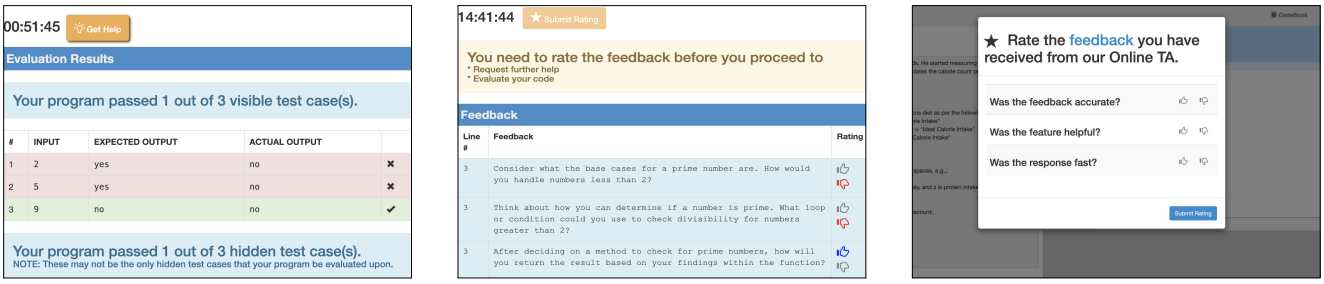


Figure 1: Workflow of randomized interventions for students

(5) **TA-Manual**: The human TAs in this group provided assistance to the students without access to any AI-generated feedback. They relied solely on their own knowledge and expertise to help the students with their assignments.
(6) **None**: This group of students did not request for feedback.

We also tracked the total number of feedback requests made by each group and their completion rates. A student is considered to have *completed* a question if their solution passes all the test cases.

System Overview. We integrated our AI Agent in Prutor [8], the students' existing web-based programming environment with a built-in autograder, to avoid impacting the user experience. When a student's code fails a test case, they can request for feedback by clicking the "Get Help" button (Figure 2a), and optionally writing in their query. Depending on the experimental group, the request for help is routed to either our AI agent or the human TAs, who are potentially augmented with AI. For each request received, the TA can see the student buggy code, failing testcases and optional student query. For AI assisted TA, draft comments are automatically generated, which they can approve, delete or modify, in addition to adding their own feedback.



(a) Students with failing tests can request help. **(b) Students receive feedback from TA or AI.** **(c) Students rate the feedback.**

Figure 2: User interface for providing hints to struggling CS1 programming students

Since student errors could span multiple lines of code, feedback could be provided for each buggy line (Figure 2b). Notably, students were unaware of the different experimental groups, and to maintain this blinding, we intentionally delayed feedback for Q1 to align with the slower response time of human TAs in Q2. Students used the provided feedback to correct their code and were required to rate each feedback line with an upvote or downvote. Additionally, they rated the overall feedback based on accuracy, helpfulness, and timeliness before proceeding with their assignment (Figure 2c). The results of these ratings are discussed in §4.4.

4 Results

In this section, we present our findings on the impact of augmenting Teaching Assistants (TA) with AI on both student outcomes and TA performance. We also present the students' perceptions of feedback quality across various experimental groups. The anonymized dataset is publicly released to aid further research [4].

4.1 Baseline Results

Our AI agent was used to service a total of 115 requests and depending on the question the feedback was sent either directly to the student or indirectly after validation by a human TA, as described in §3. We manually annotated both the AI-generated and TA created feedback for correctness. We found that the AI agent achieved a precision of 0.87 and had a hallucination rate of 8.7%. In comparison, our human TAs manually handled 17 help requests without any AI assistance, and achieved a comparable precision of 0.88. Surprisingly, there was one instance of hallucination by a human TA, i.e. the feedback was both wrong and misleading.

Our baseline analysis demonstrates that our AI agent generates feedback comparable in coverage and accuracy to human TAs. This is consistent with the results of our earlier study where we found that GPT-4 generated invalid feedback 8% of the time, hallucinated in 5% of the cases, and failed to detect 16% of the mistakes made by high school programming students [19].

As shown in Table 1, approximately 80% of students successfully completed question Q1 on average across all the groups, passing all instructor defined test cases. For question Q2, however, this completion rate dropped to 40% or lower for our experimental group of students who requested feedback. Among students who did not request any form of feedback for Q2, 56.2% completed the programming task successfully, suggesting that this group comprised of students with stronger programming skills.

We assessed the impact of each intervention on final student scores and found no statistically significant difference across groups. The median scores for all three groups in Q1 was 100/100, with a p-value of 0.96. Similarly, for Q2, although few students in the TA-Manual and TA-AI (Socratic) groups scored lower than those in the TA-AI (Default) group, this difference was not statistically significant (p-value of 0.85). In other words, our AI agent is able to generate feedback comparable to human TAs and thus *it is feasible to deploy AI-generated feedback for programming questions at scale.*

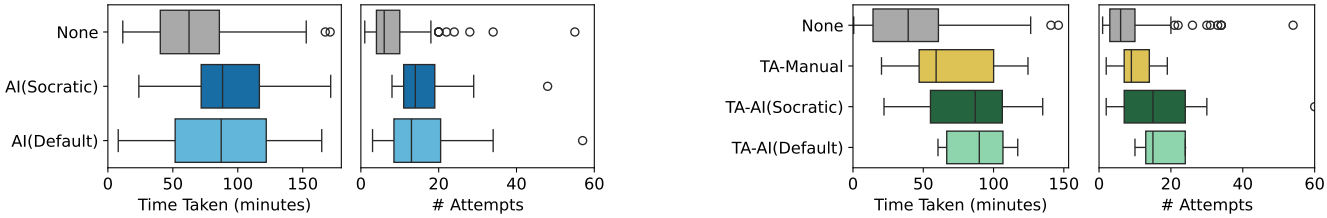
4.2 How Does AI Feedback Style Impact Student Performance?

In Figure 3a, we show the time taken and number of attempts for questions Q1. Here, *time taken* refers to the total time a student spent before their final submission, and *number of attempts* refers to the number of times they evaluated their code. It is evident that students who did not seek any feedback completed the problem faster with fewer attempts, compared to the groups who required assistance. This is not surprising since the former group was likely to comprise of stronger students. However, there were also outliers who struggled but did not seek assistance.

Among students who sought assistance, we found no significant difference in completion rates between the two styles for AI feedback in question Q1 ($p = 0.85$). However, AI (Default) group's wider inter-quartile range (IQR) suggests that *while the feedback style of AI did not matter much overall, direct feedback was more efficient for some students compared to a Socratic approach.* This is not surprising since Socratic-style feedback is less direct.

4.3 How Does Augmenting Human TAs With AI Impact Student Performance?

To understand the impact of AI augmentation on student performance, we present the student performance for Q2, which is the harder question, for three feedback approaches: traditional TA-Manual feedback and two styles of AI-augmented feedback, TA-AI (Default) and TA-AI (Socratic), in Figure 3b. We note that the AI-augmented TAs could choose to modify the AI generated feedback before sharing it with the student. Like Q1, students who did not seek help in Q2 generally performed better.



(a) Question Q1: students received AI generated hints

(b) Question Q2: hints were provided through human TAs.

Figure 3: Time taken and number of attempts across experimental groups.

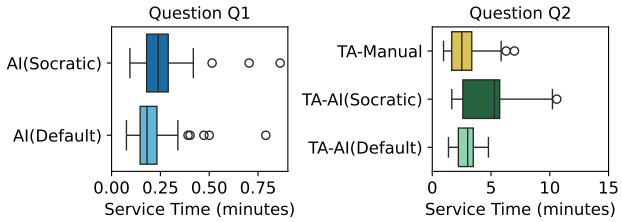


Figure 4: Feedback service time across experimental groups.

Surprisingly, the students in the TA-Manual group, where the TAs received no AI assistance, completed the task faster and with fewer attempts than the students who received AI-augmented feedback. This finding is surprising as one would expect AI-augmented TAs to perform no worse than TAs who had no AI assistance. After all, the former could ignore the AI-generated feedback and directly respond to the students in their own words. This difference was particularly pronounced in the number of attempts, with the Tukey's HSD revealing that the TA-AI (Default) group required significantly more attempts than their TA-Manual peers ($p = 0.012$).

Upon further analyzing our data, we found that human TAs working without AI assistance tended to provide shorter *concise* feedback, highlighting the immediate next step, sometimes with direct instructions for resolving programming errors, instead of hints. In other words, human TAs without AI assistance were potentially giving away the answers. *While students receiving feedback from TA-Manual were slightly faster in completing their problem, it does not necessarily translate to better learning outcomes.*

4.4 How Do Students Perceive Feedback Quality Across Different Groups?

In Figure 5, we present the students' perception of feedback accuracy and helpfulness. For Q1, AI (Default) was perceived to be slightly more accurate and helpful. For Q2, TA-AI (Default) was considered more accurate and faster, but the helpfulness rating of all three groups was similar. However, we see in Figure 3b that the students in the TA-Manual group finished faster with fewer attempts for Q2. This corroborates with previous observations that student perceptions do not always correlate with learning outcomes [1].

One potential explanation is that AI-generated feedback is very verbose, averaging 129 words per incorrect program, and tends to reveal all possible mistakes and hints to the student. On the other hand, TA-Manual feedback averages just 13 words, focusing on the immediate next step. This brevity is likely the result of the human TAs being more efficient at identifying student errors. Interestingly, we suspect some TAs from the TA-Manual group likely used ChatGPT independently to respond to student queries, because we found some unusually long (250-300 words) and grammatically flawless response for two TAs in the TA-Manual group.

In Figure 6, we see that the opinion of human experts differs from student perceptions. Not only were the experts able to identify cases of hallucinations and revealing feedback, unlike students, they rated AI (Socratic) feedback as more accurate and helpful than AI (Default) for Q1. For Q2, however, experts were aligned with students and reported that TA-AI (Default) was more accurate and helpful than both TA-AI (Socratic) and TA-Manual. In other words, *students and experts favored AI's detailed explanation, but the manual TA's brief and precise guidance was able to help students arrive at the solution faster.*

4.5 How Does AI Augmentation Impact TA Performance?

In the baseline section §4.1, we found the accuracy of LLM based assistance AI (Default) and AI (Socratic) to be comparable to that of human TA-Manual. In this section, we investigate whether a hybrid model of augmenting TAs with AI generated feedback can improve their overall performance and quality of response.

To this end, we present the feedback service time for various experimental groups in Figure 4, measured from the moment TAs began processing the student's query to when they submit their feedback. While our AI agent serviced requests in order of seconds, the three experimental groups with human TAs had response times in the range of minutes. Notably, only the TA-AI (Socratic) and TA-Manual groups showed a statistically significant difference ($p = 0.03$), with TA-Manual achieving faster response times. This finding suggests that AI assistance does not always improve the efficiency of TAs in providing feedback. In particular, the Socratic style of AI feedback appears to impose a cognitive burden on TAs, resulting in significantly longer service times compared to direct manual interventions. In other words, *providing AI feedback as reference does not necessarily improve TA efficiency. The style of the feedback also matters.*

While service time is a helpful metric for assessing TAs performance, the feedback quality is the more important metric in a pedagogical setting. To provide a comprehensive evaluation, we also manually categorized all the feedback provided during our user-study into the following qualitative metrics: accurate, helpful,

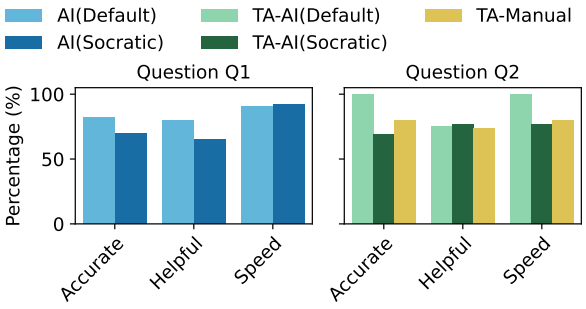


Figure 5: Feedback ratings by students.

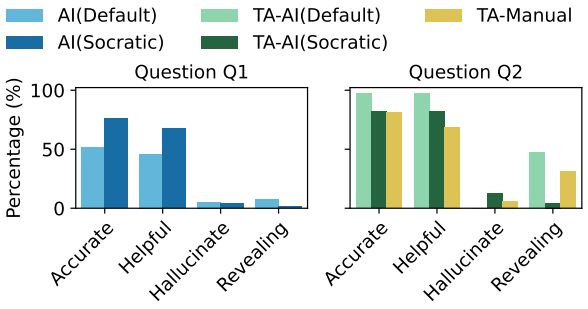


Figure 6: Feedback ratings given by expert.

hallucinate, and revealing. The summary of these metrics across different feedback groups is presented in Figure 6.

We found that TAs provided with AI assistance were found to have higher accuracy and helpfulness compared to TAs who did not have access to AI assistance. Hallucination was found to occur in the generated feedback for all the groups. The TA-AI (Default) and TA-Manual groups were found to provide feedback that was significantly more revealing compared to the Socratic style feedback, which is to be expected by design.

In about 83% of the responses, TAs enhanced the AI-generated feedback by inserting additional comments. Given that LLM generated feedback is likely to become increasingly common, we investigated whether a *human in-the-loop* can eliminate hallucination. We found that TAs never deleted AI-generated content, despite hallucination appearing in about 9% of the cases. In fact, only one TA edited the AI-generated feedback before sending it to the student. In other words, while human TAs could potentially catch gaps for an AI agent, the tendency to correct AI-generated feedback is much lower. *It is unlikely that hallucination in AI-generated feedback can be completely eliminated with a human in the loop.*

5 Limitations

Our study has some limitations. First, all of our groups were experimental groups, and we were unable to set up a control group because of fairness considerations since it was graded lab. This restriction also prevented us from conducting a Randomized Controlled Trial (RCT) where every possible group could be tested on all questions. The best we could do was to divide the students into groups between those receiving only AI hints for Q1, and those receiving TA assistance augmented by AI for Q2.

Second, our time taken metric to submit code is only an estimate, as the students could freely switch between the two questions throughout the lab. However, this variability is likely consistent across groups and should not substantially impact our conclusion. Additionally, while the time taken offers insight into student efficiency, it does not directly measure learning gains.

Our study serves as a preliminary effort to measure the impact of AI-augmented feedback in a live CS1 programming setting. Despite the promising results and relatively large size of 185 students, the small individual group size of 23 students or fewer poses a limitation, as only a small number of students requested for help. Further research with a larger sample size and diverse educational environments is necessary to validate our findings.

6 Conclusion

Our study demonstrates the feasibility of using AI to augment Teaching Assistants (TAs) in a live CS1 programming lab, representing the first large-scale real-world implementation of its kind. We found that AI-augmented TAs can deliver more accurate and helpful feedback, as perceived by both students and experts. However, this did not consistently translate into improved student performance. In contrast, traditional manual feedback by TAs was more effective overall. Contrary to our expectations, AI-augmented TAs were on average slower in providing feedback than their manual counterparts, suggesting that AI might not always improve efficiency as commonly assumed.

We identified several issues in AI generated feedback, such as verbosity, lack of context, and hallucination, which could cause confusion instead of improving learning outcomes. To address these challenges, further work is required to improve the feedback quality by generating concise, actionable feedback that preserves the pedagogical value while reducing inaccuracies.

The implications of our study extend beyond the CS1 classroom. As AI tools in education become more ubiquitous, there is a pressing need to carefully evaluate their impact and design better interfaces for both TAs and students to effectively leverage these tools. Our findings highlight the potential pitfalls of over-reliance on AI, which should be used as a tool to support, and not replace, human educators. Future research should focus on developing integrated approaches that enhance, rather than undermine, the teaching process, ultimately working towards the goal of improving educational access and quality at scale.

Acknowledgment

We express our gratitude to Debapriya Basu Roy and Meeta Bagga, without whose support this user study would not have been possible, and to the CS1 students and teaching assistants for their active participation. Additionally, we would like to thank the anonymous reviewers for their valuable feedback. This research is supported by the National Research Foundation Singapore under the AI Singapore Programme (AISG Award No: AISG2-TC-2023-009-AICET).

References

[1] Abednego Abednego, Sefnath Nuniary, Emma Rumahlewang, and John Rafafy Batlolona. 2023. The Correlation between Student Perception and Learning Motivation: Blended Learning Strategy. *AL-ISHLAH: Jurnal Pendidikan* 15, 2 (2023), 1338–1346.

[2] Umair Z Ahmed, Zhiyu Fan, Jooyong Yi, Omar I Al-Bataineh, and Abhik Roychoudhury. 2022. Verifix: Verified repair of programming assignments. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 31, 4 (2022), 1–31.

[3] Umair Z Ahmed, Nisheeth Srivastava, Renuka Sindhgatta, and Amey Karkare. 2020. Characterizing the pedagogical benefits of adaptive feedback for compilation errors by novice programmers. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training*. 139–150.

[4] AI Centre for Educational Technologies (AICET). 2024. User study artifacts containing CS1 programming assignments and experimental results. https://github.com/ai-cet/sigcse2025-userStudy-artifacts.

[5] John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. 1995. Cognitive tutors: Lessons learned. *The journal of the learning sciences* 4, 2 (1995), 167–207.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[7] Tracy Camp, Stu Zweben, Ellen Walker, and Lecia Barker. 2015. Booming enrollments: Good times?. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. 80–81.

[8] Rajdeep Das, Umair Z Ahmed, Amey Karkare, and Sumit Gulwani. 2016. Prutor: A system for tutoring CS1 and collecting student programs for analysis. *arXiv preprint arXiv:1608.03828* (2016).

[9] Paul Denny, Stephen MacNeil, Jaromir Savelka, Leo Porter, and Andrew Luxton-Reilly. 2024. Desirable characteristics for ai teaching assistants in programming education. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*. 408–414.

[10] Sumit Gulwani, Ivan Radicek, and Florian Zuleger. 2018. Automated clustering and program repair for introductory programming assignments. In *Proceedings of the 39th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*. 465–480.

[11] James A Kulik and John D Fletcher. 2016. Effectiveness of intelligent tutoring systems: a meta-analytic review. *Review of educational research* 86, 1 (2016), 42–78.

[12] Mark Liffiton, Brad E Sheese, Jaromir Savelka, and Paul Denny. 2023. Codehelp: Using large language models with guardrails for scalable support in programming classes. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*. 1–11.

[13] Julia M Markel, Steven G Opferman, James A Landay, and Chris Piech. 2023. Gpteach: Interactive ta training with gpt-based students. In *Proceedings of the tenth acm conference on learning@ scale*. 226–236.

[14] OpenAI. 2023. New Models and Developer Products Announced at DevDay. https://openai.com/blog/new-models-and-developer-products-announced-at-devday Accessed: 2024-07-22.

[15] Andrew Petersen, Michelle Craig, Jennifer Campbell, and Anya Tafliovich. 2016. Revisiting why students drop CS1. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*. 71–80.

[16] Tung Phung, Victor-Alexandru Padurean, José Cambronero, Sumit Gulwani, Tobias Kohn, Rupak Majumdar, Adish Singla, and Gustavo Soares. 2023. Generative AI for Programming Education: Benchmarking ChatGPT, GPT-4, and Human Tutors. *CoRR* abs/2306.17156 (2023). https://doi.org/10.48550/arXiv.2306.17156 arXiv:2306.17156

[17] Tung Phung, Victor-Alexandru Pădurean, Anjali Singh, Christopher Brooks, José Cambronero, Sumit Gulwani, Adish Singla, and Gustavo Soares. 2023. Automating Human Tutor-Style Programming Feedback: Leveraging GPT-4 Tutor Model for Hint Generation and GPT-3.5 Student Model for Hint Validation. *arXiv preprint arXiv:2310.03780* (2023).

[18] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950* (2023).

[19] Shubham Sahai, Umair Z Ahmed, and Ben Leong. 2023. Improving the Coverage of GPT for Automated Feedback on High School Programming Assignments. In *NeurIPS'23 Workshop Generative AI for Education (GAIED). MIT Press, New Orleans, Louisiana, USA*, Vol. 46.

[20] Linda J Sax, Kathleen J Lehman, and Christina Zavala. 2017. Examining the enrollment growth: Non-CS majors in CS1 courses. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. 513–518.

[21] Jodi L Tims, Cindy Tucker, Mark A Weiss, and Stuart Zweben. 2023. Computing Enrollment and Retention: Results from the 2021–22 Undergraduate Enrollment Cohort. *ACM Inroads* 14, 4 (2023), 24–43.

[22] Kurt VanLehn. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational psychologist* 46, 4 (2011), 197–221.

[23] Sierra Wang, John Mitchell, and Chris Piech. 2024. A large scale RCT on effective error messages in CS1. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. 1395–1401.

[24] Christopher Watson and Frederick WB Li. 2014. Failure rates in introductory programming revisited. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*. 39–44.

[25] Jooyong Yi, Umair Z Ahmed, Amey Karkare, Shin Hwei Tan, and Abhik Roychoudhury. 2017. A feasibility study of using automated program repair for introductory programming assignments. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (FSE)*. 740–751.

[26] JD Zamfirescu-Pereira, Laryn Qi, Björn Hartmann, John DeNero, and Narges Norouzi. 2024. 61A-Bot: AI homework assistance in CS1 is fast and cheap–but is it helpful? *arXiv preprint arXiv:2406.05600* (2024).