

Enhancing active learning through collaboration between human teachers and generative AI

Kritish Pahi^{1,a}, Shiplu Hawlader^{1,a}, Eric Hicks^a, Alina Zaman^a, Vinhthuy Phan^{a,*}

The University of Memphis, Memphis, TN 38152 United States

ARTICLE INFO

Keywords:

Cooperative/collaborative learning
Architectures for educational technology system
Improving classroom teaching

ABSTRACT

To address the increasing demand for AI literacy, we introduced a novel active learning approach that leverages both teaching assistants (TAs) and generative AI to provide feedback during in-class exercises. This method was evaluated through two studies in separate Computer Science courses, focusing on the roles and impacts of TAs in this learning environment, as well as their collaboration with ChatGPT in enhancing student feedback. The studies revealed that TAs were effective in accurately determining students' progress and struggles, particularly in areas such as "backtracking", where students faced significant challenges. This intervention's success was evident from high student engagement and satisfaction levels, as reported in an end-of-semester survey. Further findings highlighted that while TAs provided detailed technical assessments and identified conceptual gaps effectively, ChatGPT excelled in presenting clarifying examples and offering motivational support. Despite some TAs' resistance to fully embracing the feedback guidelines—specifically their reluctance to provide encouragement—the collaborative feedback process between TAs and ChatGPT improved the quality of feedback in several aspects, including technical accuracy and clarity in explaining conceptual issues. These results suggest that integrating human and artificial intelligence in educational settings can significantly enhance traditional teaching methods, creating a more dynamic and responsive learning environment. Future research will aim to improve both the quality and efficiency of feedback, capitalizing on unique strengths of both human and AI to further advance educational practices in the field of computing.

1. Introduction

The interest in Artificial Intelligence (AI) has surged among the general public due to its transformative role in various sectors and daily life. This surge is manifested in the increasing demand for AI education from students of diverse academic backgrounds, each drawn by the potential of AI to innovate across fields and offer promising career paths. However, the diversity in students' preparation and understanding of AI poses significant challenges for educators tasked with delivering effective AI education to such a heterogeneous group.

Achieving AI Literacy for All necessitates addressing several key challenges: how to design learner-centered technologies that foster understanding of AI, ensuring accessibility to quality AI education for diverse student populations, accommodating varied learning paces and styles, and providing personalized and engaging learning experiences. Traditional approaches, often characterized by a one-size-fits-all

methodology, are insufficient in meeting these demands, highlighting the need for innovative pedagogical strategies.

To bridge the gap in AI literacy, it is imperative to shift from traditional pedagogical models towards more inclusive and adaptive teaching methods. Personalized instruction emerges as a critical solution to cater to the individual needs and aptitudes of students, thereby mitigating the risk of incomprehension and the expertise reversal effect. Active learning further complements this by positioning students as engaged participants in their learning process, enabling hands-on experiences that facilitate the application of theoretical knowledge. Additionally, early intervention plays a vital role in supporting students who might otherwise fall behind.

In response to these challenges, building on previous research on active learning approaches that enable teachers and teaching assistants (TAs) to help students during in-class coding exercises [1–6], in this work, we propose an approach that aims to provide active learning in a

* Corresponding author.

E-mail addresses: kpahi@memphis.edu (K. Pahi), mhwlader@memphis.edu (S. Hawlader), elhicks@memphis.edu (E. Hicks), azaman@memphis.edu (A. Zaman), vphan@memphis.edu (V. Phan).

¹ These authors have equal contribution to this work as first author.

way that accommodate learners with different abilities and styles. This approach integrates active learning with personalized instruction, utilizing the guided collaborative efforts of TAs and generative AI tools, such as ChatGPT, to enhance the learning experience. Our approach is built on four pillars: promoting an active learning environment through in-class exercises, fostering personalized learning through continuous progress monitoring by TAs and ChatGPT, offering immediate feedback for timely intervention, and synergizing TAs and generative AI to optimize the learning process. We aim to address four specific research questions:

- RQ1 How effectively can TAs identify and assist struggling students on the hybrid platform?
- RQ2 Can TAs recognize common challenges or patterns during exercises, and what are these challenges or patterns?
- RQ3 How do the quality, depth, and timeliness of feedback from human TAs compare with that from ChatGPT?
- RQ4 What strategies best integrate the strengths of human TAs and ChatGPT to improve feedback accuracy and usefulness in coding exercises?

To address these questions, two studies were conducted in several undergraduate and graduate courses in the Computer Science and Data Science programs at a medium-size university. In the first study, we found that the course TA was able to detect 32 instances where students experienced difficulties during 10 different exercises. The TA also identified that 4 common challenges that students faced. Through this assistance, the TA helped to *backtracking* was the most challenging topic, with which many students had troubles. We also found that the instructor agreed with the TA's assessment more than 99% of the time. An end-of-semester survey reveals suggest that almost all students were eager to participate in in-class exercises and enjoyed the active learning experience. In the second study, we found that TAs and ChatGPT excelled in different areas of feedback provision. Further, the collaboration between TAs and ChatGPT led to significant improvements in feedback quality in several areas.

The proposed framework and approach have potentials to create a synergistic collaboration between human TAs and AI. The outcome can lead to a scalable solution that make education more inclusive and personalized.

2. Background

Challenges of AI Literacy for All: The concept of "literacy" traditionally pertains to proficiency in using written language for expression and communication. With the integration of AI into the educational sphere globally, the definition of AI literacy has expanded to encompass various aspects. These include understanding AI's fundamental operations and the ethical use of AI applications in daily life [7,8]. Additionally, AI literacy involves the application of AI knowledge, concepts, and applications across diverse scenarios [9–11], along with the ability to evaluate, critique, and design AI applications [8,9,12,13]. An important aspect of AI literacy also includes recognizing the principles of fairness, accountability, transparency, and ethics in AI technologies [9,14,15].

Addressing the challenges inherent in AI education necessitates the development of technologies focused on the learner, incorporating all the aforementioned principles. Achieving comprehensive AI literacy education presents significant obstacles, among which ensuring inclusivity remains paramount [9]. Furthermore, facilitating effective interactions between humans and AI agents has emerged as a considerable challenge [10,16], particularly in the context of establishing AI teaching standards for K-12 education [17]. Tackling these challenges is crucial for adequately preparing students to thrive in and contribute meaningfully to an AI-centric future.

Personalized Learning: Personalized learning aims to personalize the learning experience according to the needs, goals, and skills of students

[18–20]. Personalized learning encompass a broad range of possibilities-from customized interfaces to adaptive tutors, from student-centered classrooms to learning management systems-that expectations run high for their potential to revolutionize learning [21]. Without such an approach, there's a risk of fostering incomprehension and the *expertise reversal effect* among a diverse student population [22].

In the context of fostering AI Literacy for All, personalized learning emerges as a pivotal strategy to democratize access to AI education, catering to the unique needs, preferences, and learning styles of each student. This approach is instrumental in making AI concepts accessible and engaging to a broad spectrum of learners, thereby enhancing overall engagement by aligning educational content with students' interests and capabilities [18,19,23–27]. Offering personalized pathways prevents students from falling behind, keeps them engaged in learning, and increases their confidence in their own abilities [18,19].

Moreover, personalized learning plays a crucial role in promoting self-regulation and independence among students, encouraging them to take an active role in their AI learning journey. By setting personal goals and engaging in reflective practices, students develop a sense of ownership over their learning, which is essential for navigating the complex and rapidly evolving field of AI. Furthermore, the inclusivity of personalized learning ensures equitable access to AI education for students from diverse backgrounds, including those who are traditionally underrepresented or at-risk in the field of technology [25].

The importance of personalized learning is underscored by research indicating increased student satisfaction and confidence, which are key to fostering a positive learning environment and preparing students for the challenges of a future where AI plays a significant role. A comprehensive review of personalized learning strategies and their impact on educational outcomes is provided by Zhang et al. [28]. By implementing personalized learning approaches, educators can make significant strides in achieving AI Literacy for All, ensuring that students are not only consumers of AI technology but also informed participants in its development and application.

Active Learning and Active Learning Environments: Active learning, a pedagogical approach that engages students directly in their educational journey, plays a critical role in enhancing AI Literacy for All. This method, which includes problem-solving, discussions, and hands-on activities, fosters critical thinking and practical application of concepts, aligning with educational psychology and pedagogical research such as Vygotsky's Zone of Proximal Development [29,30], constructivist learning theories [31,32], and cognitive load theory [33–35]. The efficacy of active learning in increasing motivation, retention, and practical understanding of concepts is well-supported [6,36–39], making it particularly relevant for AI education.

In the context of AI education, tools that facilitate active learning through in-class exercises are invaluable. For instance, Codeopticon offers a real-time interface that allows tutors to monitor students' coding activities and provide instant feedback, crucial for AI programming tasks [40]. VizProg enhances this by enabling instructors to visualize students' coding progress in real-time through a 2D Euclidean map, revealing their problem-solving strategies and advancements [41]. Furthermore, Code4Brownies encourages teaching assistants to engage directly in in-class exercises, supporting instructors in delivering a more interactive and supportive learning experience [1,3,5,6]. These tools exemplify how active learning strategies can be effectively applied to AI education, facilitating a deeper, more accessible understanding of AI concepts for students from diverse backgrounds, thereby advancing the goal of AI Literacy for All.

Early intervention: Early intervention plays a crucial role in promoting inclusiveness in education, offering essential support to struggling and underrepresented students in large courses [42–45]. This approach enables the identification of students who need special attention early on, facilitating tailored instructional strategies to meet individual needs [3]. By monitoring students' interactions with AI-related course materials and their communication with instructors, educators can adjust teaching

methods to enhance understanding and engagement with AI concepts.

Metrics such as login frequency, material access, discussion participation, and total time spent on course platforms serve as indicators of student engagement, which is closely linked to performance [46]. In the context of AI education, leveraging these metrics allows for a more responsive and personalized learning experience. Early intervention thus not only addresses immediate learning challenges but also ensures that all students, regardless of their starting point, can achieve AI literacy, making this approach a key strategy in broadening access to AI education and empowering students to effectively engage with and contribute to the AI field.

The Role of Teaching Assistants (TAs): TAs are pivotal in STEM and AI education, particularly in facilitating active and personalized learning. Their role is critical in shaping the learning environment and influencing both student outcomes and their journey towards AI literacy [47–50]. However, the challenge arises when many TAs, despite their content knowledge, face a lack of pedagogical training. This deficit is especially pronounced in computer science and AI education, where clarifying misconceptions and implementing active learning strategies are paramount for building foundational AI competencies [51–53].

To maximize the benefits of TAs in AI education while mitigating these challenges, innovative strategies have been introduced. For instance, Code4Brownies [1] promotes the involvement of TAs in in-class exercises, supervised by course instructors, to ensure students receive the support they need. This collaboration is especially beneficial in data science and machine learning courses, where TAs can utilize instructional cloud-based technologies to facilitate hands-on learning experiences [54]. Furthermore, enhancing TA feedback on coding assignments, as explored in [5], emphasizes the critical role of TAs in improving students' understanding and application of AI concepts [55].

By providing TAs with necessary instructional knowledge and skills as well as actively involving them in the AI learning process, we can create a more supportive and effective educational environment. This approach not only leverages TAs' potential to contribute to AI literacy but also ensures that all students, regardless of their background, have equitable access to quality AI education.

AI in Education: The integration of AI into educational settings presents both significant opportunities and complex challenges. AI technologies, including software tools and chatbots, have been developed to make computing accessible for younger learners and to enrich the learning experience across various educational levels. For instance, "Machine Learning for Kids" offers interactive online demos that allow students to train classifiers, introducing them to core concepts of AI without overwhelming complexity. Similarly, "Calypso for Cozmo" integrates speech recognition and state machine programming, providing a hands-on approach to understanding AI's workings [17]. These examples illustrate the specific ways AI tools are being used to demystify programming and machine learning for students.

However, the expansion of AI chatbots like ChatGPT into education, ranging from offering instructional support to assisting with essay writing, introduces challenges such as misinformation and plagiarism [56,57]. While these tools promise to revolutionize how students learn by providing personalized assistance and feedback, educators must navigate the ethical implications, including concerns over bias and privacy [58,59]. The adaptability of ChatGPT across academic domains is noteworthy, yet its utility hinges on careful integration, planning, and oversight by educators to ensure these tools are used responsibly and effectively [60,61].

The integration of ChatGPT into educational platforms, such as ChatTutor and LingoBot, demonstrates its potential to personalize tutoring and language learning. ChatTutor² employs ChatGPT to adapt tutoring sessions to the individual needs of students, while LingoBot³

uses it to create immersive language learning experiences. These specific applications show how ChatGPT can be tailored to diverse educational contexts, enhancing both engagement and understanding [62].

Moreover, AI tools like Duolingo's adaptive language learning software and Gradescope's automated grading system exemplify the transformation of classrooms by personalizing instruction and providing instant, detailed feedback. These innovations address critical challenges such as unequal access to quality education, showcasing Knowji's vocabulary app as a tool that customizes study plans for students in under-resourced communities, thereby promoting inclusivity [63]. However, the deployment of AI in educational contexts, such as Ivy.ai's university chatbots, raises important questions about data privacy and the potential for algorithmic bias. The responsible implementation of AI tools, underscored by Amplify's personalized learning platform, emphasizes the need for educators to critically evaluate and thoughtfully integrate AI technologies into learning environments⁴.

3. An Approach to AI-assisted Instruction

Our approach to promote AI Literacy for All envisions a modern classroom that combines active learning and personalized learning, augmented by teaching assistants (TAs) and generative AI tools, such as ChatGPT. This collaborative team enhances an instructor's ability to conduct in-class exercises, while aiding students in both completing and understanding exercises. The approach aims to promote an active learning environment for experiential learning, to foster personalized learning by monitoring individual student progress, to facilitate immediate feedback for timely intervention.

The learning environment in Fig. 1, depicts a classroom that aims to accommodate the goal of AI Literacy for All. The hybrid environment consists of two key components:

1. **The physical classroom** with students, TAs, instructors, and their computers.
2. **The virtual space** which facilitates active learning through in-class exercises. This component consists of three modules: (i) Student module, (ii) Teacher/TA module, and (iii) Generative AI (e.g. ChatGPT) module. Each module interacts with each role, and facilitates communications within the learning environment.

Teacher Module: empowers instructors and TAs to effectively manage in-class exercises during lectures. It provides real-time access to student work, enabling instructors and TAs to grade submissions and provide feedback. Although instructors and TAs technically serve as "teachers," their practical roles differ. Instructors are primarily responsible for delivering lectures and leading discussions. In contrast, TAs focus on silently monitoring student progress during exercises. This allows TAs to give targeted feedback to students and alert instructors to specific needs, such as students requiring additional support or identifying common technical or conceptual challenges encountered by a majority of students during an exercise. Teachers can also communicate with generate AI, namely ChatGPT, to get a second opinion on student work.

Student Module: helps engage students in in-class exercises and facilitate virtual interactions with instructors and TAs. It allows students to submit their code for review and seek assistance. Students may also receive proactive feedback from instructors or TAs, who monitor their progress and intervene as necessary to provide needed guidance. This direct communication with instructors and TAs occurs within the virtual space. In contrast, interactions with generative AI are mediated; instructors and TAs serve as conduits, curating and refining AI-generated feedback before it reaches students. This intermediary role ensures that the feedback from AI is contextually relevant, accurate, and

² <https://chattutor.io/>

³ <https://lingobot.chat/>

⁴ <https://amplify.com/>

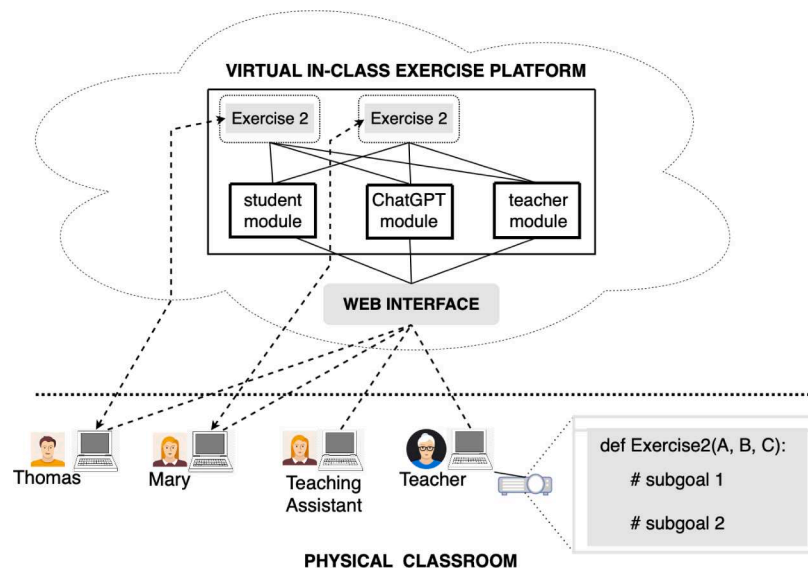


Fig. 1. An active-learning environment, where generative AI (e.g. ChatGPT), instructors, and teaching assistants collaborate to assist students with in-class coding exercises.

educationally sound.

ChatGPT Module: facilitates interaction with OpenAI's ChatGPT to support TAs in assisting students. Unless specified otherwise, ChatGPT is supplied with and instructed to follow a set of guidelines for providing effective feedback. For a discussion on effective feedback principles, see [5]. These guidelines encompass:

1. **Positive Reinforcement:** The feedback consistently encourages students, highlighting specific positive actions they have taken in their coding process.
2. **Avoid Revealing Solutions:** The module guides students toward finding solutions on their own, offering general strategies or hints without providing direct code corrections or detailed implementation instructions.
3. **Clear Actionable Steps:** Feedback includes clear, actionable steps, suggesting specific tasks or coding constructs for the students to use or consider next.
4. **Complete Sentence Structure:** The feedback is structured in well-formed, complete sentences, enhancing clarity and understanding.
5. **Pinpoint Knowledge Gaps:** The module identifies and articulates specific areas where a student's understanding may be lacking, focusing on clarifying issues rather than providing explicit solutions.
6. **Utilize Examples for Testing:** Feedback includes variable values or output examples, enabling students to test their own logic and understanding through practical application.

This structured approach provides ChatGPT as much information as possible to aid student learning.

3.1. Two Implementations of The Proposed Educational Environment

Based on the architecture shown in Fig. 1, we developed two distinct systems, which can effectively support AI Literacy. Both have been used in undergraduate and graduate courses in the Computer Science programs, and the Data Science program at our university. The first system caters to advanced courses like AI, machine learning, and data science. The second system is tailored for problem-solving and algorithmic courses, including CS0, CS1, CS2, and Algorithms. This differentiation is crucial to address the unique pedagogical needs of each domain. Coding, a key component of AI Literacy, requires distinct teaching approaches and environments. While coding courses focus on foundational programming skills and algorithmic problem-solving, AI/ML/DS courses

involve advanced topics such as data handling and model building. These separate systems ensure that each area's specific educational requirements are met, enhancing the overall effectiveness of AI Literacy education.

The first system, detailed in [54], implements the architecture using Jupyter Lab and JupyterHub. A Go language-based server manages module communications, optimized for efficient networking. This configuration facilitates interactive learning through the use of Jupyter notebooks, which combine code and text. Jupyter notebooks are especially beneficial in AI and Data Science courses for their adaptability and ease of use [64]. Utilizing Jupyter Lab, within the Anaconda Python distribution, simplifies access to complex libraries such as sklearn and scipy, eliminating the need for complicated installation processes. Fig. 2 shows the student interface's which communicate with the Student Module (see Fig. 1); It shows a student working on an exercise on a Jupyter notebook in a Data Science course. Via a set of buttons embedded on Jupyter notebooks, students can ask for help, share their work, and get feedback from TAs and the course instructor.

The second system, detailed in [1], also follows the architecture, but employs an integrated development environment (IDE) instead of Jupyter notebooks. This choice of using an IDE, specifically SublimeText, is particularly apt for teaching introductory programming courses. IDEs are more suitable than Jupyter notebooks for these courses because they not only provide a more realistic programming environment that students are likely to encounter in professional settings, but they also offer comprehensive coding tools for writing, debugging, and testing code. While Jupyter notebooks are excellent for interactive exploration and data analysis, IDEs offer a more robust and versatile platform for learning fundamental programming skills, including the intricacies of debugging. This makes IDEs a more appropriate choice for introductory programming courses, as reflected in the successful application of this system in such courses. Fig. 3 shows the student interface's which communicate with the Student Module (see Fig. 1); It shows a student working on an exercise on a coding exercise on an IDE. Via the IDE's menu, students can ask for help, share their work, and get feedback from TAs and the course instructor.

4. Methodology

This research comprises two studies that explore the dynamic roles of TAs and the synergistic interactions of TAs with ChatGPT within the proposed educational framework.

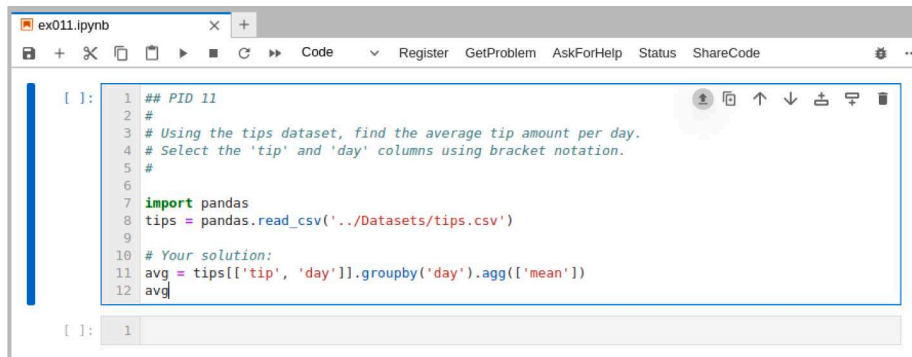


Fig. 2. System 1: a student view showing a student working on an in-class exercise on a Jupyter notebook. Students can share their work, ask for help, and receive feedback from TAs and course instructors.

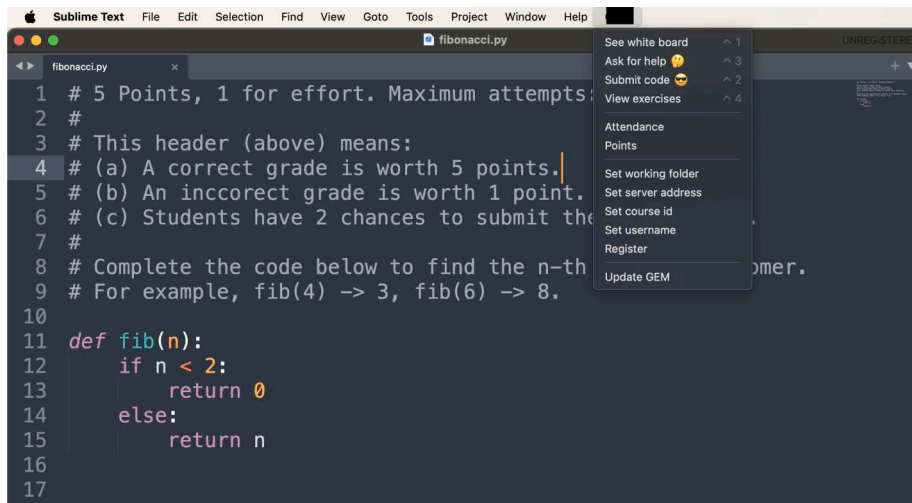


Fig. 3. System 2: student working on an in-class exercise in an IDE (Sublime Text). Students can share their solution, ask for help, and receive feedback from TAs and course instructors using the plugin as shown.

4.1. Study 1: Effectiveness of TAs in the Learning Environment

To investigate the role of TAs in the learning environment, we conducted a study focusing on two key research questions:

RQ1 How effectively can TAs identify and assist struggling students on the hybrid platform?

RQ2 Can TAs recognize common challenges or patterns during exercises, and what are these challenges or patterns?

Study Design: The study focused on the TA's effectiveness in identifying students facing difficulties and common patterns in their work during in-class exercises. TA involvement in the in-class exercises begins after the instructor publishes the problem for students to work on. As soon as the students download the problem from the server to their Jupyter notebooks, TAs can monitor their work in a grid style view. TAs actively keep track of student progress through the interface which displays their last active time. For simplicity, the green color signifies students who are actively working on the problem while the red color signifies students are stuck in their work and haven't make any progress for the last 5 minutes. To help such struggling student groups, TAs can directly access their working space to observe their progress and provide appropriate in-line feedback as well as generic feedback on their code. On the other hand, students can view that feedback via the status page which is only accessible to them. This mechanism provides timely help to students who are struggling and unable to progress in their work.

Furthermore, with our implementation, TAs can mark selective students with a Watched list. Typically, TAs add students to the watch list who need attention from the instructor. Students who are going off-track from the target solution or whose solutions need to be discussed are common additions to the Watch List. As the Watch List is accessible to the instructor, they can discuss such student's code with the class anonymously. This scenario is equivalent to a 1-on-1 discussion between the instructor and a student who needs the most help in class.

After the lecture when the in-class sessions end, the TA reviewed all the complete and partial solutions from the students. The goal was to identify common challenges or patterns in their solutions. The TA assessed the students' code using 4 labels: Adequate, In Progress, No Progress, and Incorrect. The labels were then presented to the instructor for validation.

At the end of the semester, a survey was conducted to assess student perceptions of the instructional approach and its impact on their learning experience.

Participants: The study involved 26 graduate students, predominantly from international background, enrolled for a graduate-level Algorithm course during the Summer Term of 2023. The class comprised of 16 males and 10 females, all of whom had prior exposure to introductory programming courses. The course instructor had taught this course multiple times over the course of 19 years and the TA for this course was a doctoral student at the university. Students were briefed about the in-class exercises contributing to their grades for the class.

Classroom Setting: The course combined traditional lectures with in-

class exercises, using Jupyter Notebook for both delivery and student engagement. The instructor used the first system (described in section 3), which implements the architecture depicted in Fig. 1. In-class exercises reinforced lecture topics, with students encouraged to share their work with the instructor and TAs. Exercises lasted between 10-15 minutes. The TA monitored student progress through a web interface, graded, and provided feedback on submissions alongside the instructor. A total of 38 exercises were administered covering various topics (Table 1).

Data Collection: The data collection unfolded in three phases. In the first 3 weeks, only the instructor managed monitoring, grading, and feedback, with 22 exercises resulting in 568 student submissions. In the next 3 weeks, we explored the collaborative dynamics of the TA's involvement, featuring 12 exercises and 235 submissions jointly addressed by the instructor and TA. The TA analyzed both complete and incomplete student work outside class hours, identifying common challenges. Finally, survey data was collected at the end of the semester.

4.2. Study 2: Potentials for Collaboration Between TAs and Generative AI

This study investigates the potential for collaboration between TAs and generative AI, specifically focusing on the quality of feedback in 4 different scenarios when human TAs and ChatGPT worked alone and when they collaborated. The research questions are:

RQ3 How do the quality, depth, and timeliness of feedback from human TAs compare with that from ChatGPT?

RQ4 What strategies best integrate the strengths of human TAs and ChatGPT to improve feedback accuracy and usefulness in coding exercises?

Study Design:

TAs and ChatGPT were asked to provide feedback to student work on in-class exercises. Four different scenarios were investigated:

1. **Uninfluenced Feedback:** TAs and ChatGPT provide feedback to student work. Both are provided with the problem descriptions and learning objectives. This scenario captures the cases when TAs are not provided any special training to do their jobs [55]. Although ChatGPT is trained extensively by OpenAI to understand code, with respect to the course, it is not provided additional information or goals other than the aforementioned information.
2. **Guided Feedback:** Both TAs and ChatGPT are provided with the same guideline on how feedback to students should be provided. The guideline is extracted from research on effective feedback provision [5]. Effective feedback, described below, emphasizes constructive criticism, actionable insights, closing gaps in understanding, and clarity.
3. **Feedback Review by TAs:** After ChatGPT provides *uninfluenced feedback* (Scenario 1 above), TAs review and revise ChatGPT's feedback. TAs are provided with a guideline on effective feedback provision before reviewing and revising ChatGPT's feedback. This scenario explores the potentials for ChatGPT and TAs to work as a team to provide feedback. This scenario is particularly applicable to

large classes, in which one or two TAs are not able to review code and provide feedback to a large number of students in a short amount of class time.

4. **Feedback Review by ChatGPT:** After TAs review and revise ChatGPT's feedback of a student's work (Scenario 3), ChatGPT reviews and revises the TAs' feedback. ChatGPT is provided with guidelines on effective feedback provision. This scenario similarly explores the potential for ChatGPT and TAs to work as a team to provide feedback. This scenario explores the potential for ChatGPT, equipped with a guideline for effective feedback provision, to improve upon the TAs' feedback by aligning it with the guideline.

Our criteria for effective feedback were developed through a literature review and previous studies [5,55,65–71]. Seven criteria for effective feedback include:

- **Technical Quality:** Technical correctness is central to any feedback, especially in coding exercises. Feedback should not mislead or misinform. Any suggestions, corrections, or advice given should be technically accurate and feasible.
- **Encouragement:** Feedback should motivate and encourage the learner. While pointing out errors or areas of improvement, it's vital to ensure the tone remains positive and uplifting, fostering a growth mindset.
- **Solution Concealment:** It's important that feedback guides students towards an answer rather than directly offering it. This ensures that the student engages with the problem-solving process and internalizes the concept.
- **Identifying Next Actions:** Feedback should not just point out what's wrong, but also suggest what to do next. Clear directions or hints about how to address the issue can effectively guide students.
- **Clarity and Completeness:** Using complete sentences ensures clarity and reduces the chances of misinterpretation. Concise yet clear feedback aids comprehension.
- **Gap Identification:** The feedback should highlight the discrepancy between the current solution and the expected solution. This helps students understand precisely where their understanding or approach deviates from the expected norm.
- **Use of Examples:** Whenever possible, feedback should include examples to clarify a point or provide a more concrete understanding of the concept or error at hand.

Scoring feedback: To quantitatively assess the feedback based on the aforementioned criteria, each feedback message from ChatGPT or a TA is rated with a score of 0, 1, or 2 for each of the evaluation criteria. Specifically:

Score 0 - Absent: The criterion is entirely missing from the feedback. For example, if there's no encouragement or positive reinforcement, this score is given.

Score 1 - Present: The feedback includes the criterion, but its implementation may be cursory, incomplete, or not of the highest quality.

Score 2 - High Quality: The criterion is not only present in the feedback but is also implemented in a comprehensive and high-quality manner. This score represents the ideal state for each criterion.

Participants: A total of 87 students participated in the study, drawn from two distinct courses. 44 undergraduate students were enrolled in Spring 2022's CS1 offering, while the remaining 43 participants were enrolled in a graduate-level Algorithms course held in Summer 2023. The student demographics reflected the course levels, with CS1 primarily attracting undergraduate U.S. nationals and the Algorithms course composed predominantly of international graduate students.

As with the first study, experienced instructors and TAs were involved in these two courses. Moreover, we recruited two additional TAs to assess student code and feedback. Both of them were graduate students and one of them was an international student.

Table 1
In-class exercises in Study 1.

Category	Number of exercises
Iterative design	3
Running time analysis	11
Recursive design	9
Repeated substitution	2
Binary search	3
Sorting	2
Backtracking problem	8
Total	38

Classroom Setting: Similar to the first study, these courses also combined the traditional lectures with in-class exercises using Sublime Text. Exercises lasted between 10-15 minutes. The instructor used the second system described in Section 3 to conduct in-class exercise sessions, which included broadcasting exercises, students asking for help, instructor/TAs providing feedback, submitting code for grading, etc. The exercises covered algorithmic topics, ranging from basic CS1 concepts like Function Basics and Lists to advanced topics like Recursion and Dynamic Programming in Algorithms.

Data Collection: Student work on in-class exercises was collected in two undergraduate courses. In CS1, a total 22 in-class exercises were conducted, resulting in 746 submissions. Among them 202 were incorrect, 282 were correct, 259 were auto-graded, and 3 remain ungraded. In Algorithms, 38 in-class exercises were conducted which resulted in a total of 803 submissions. TAs and ChatGPT were asked to provide feedback on a total of 30 randomly chosen submissions (21 from CS1 and 9 from Algorithms).

5. Results

To assess the utility of the proposed approach to conducting active learning in a generative-AI assisted environment, we conducted two studies as described in Sections 4.1 and 4.2. Below, we report the findings on the research questions.

5.1. TAs' Identification of Struggling Students and Common Difficulties

In Study 1, we set out to assess TA's effectiveness in identifying struggling students as well as common difficulties students exhibit during in-class exercises. Over the course of 10 different exercises, the course TA identified 32 students struggled and placed them on a Watch list to notify the instructor; see Table 2. In most exercises, fewer than 4 students were struggling. But in some exercises, up to 40% of participating students struggled. The most challenging topic was *backtracking*, where a number of students struggle in 6 exercises.

An analysis of TA's comments on student code showed that the TA able to detect common difficulties that students were having in 4 out of 38 exercises. Specifically:

- On the topic of Running time analysis (exercise 27), 14 out of 25 students did not follow the strategy that they were supposed to follow.
- On the topic of Backtracking problem (exercise 39), based on the lack of submission, it seems that the majority of students made no progress.
- On the topic of Backtracking problem (exercise 43), a noticeable number of student failed to implement an iterative solution to verify duplicates in a list.
- On the topic of Backtracking problem (exercise 45), most students either made no progress or submitted incomplete solutions.

Table 2

In 10 exercises where the TA assisted to identify struggling students, the TA identified 32 students, who struggled during in-class exercises.

Exercise	Students on Watch List	Participation	Problem Category
1	3	26	Iterative design
2	1	24	Sorting
3	1	19	Recursive design
4	1	22	Recursive design
5	6	15	Backtracking
6	6	19	Backtracking
7	1	12	Backtracking
8	1	19	Backtracking
9	8	23	Backtracking
10	4	17	Backtracking

This finding underscores a strong possibility that the subject of backtracking was most challenging. Many students had common difficulties across multiple in-class exercises.

5.2. Validation of TAs' Assessment

To validate the TA's assessment of student work, the instructor went through the TA's evaluation of each student submission and determined if he agreed with the TA's assessment. For each student submission, the TA assigned one of these labels: Adequate, No progress, In progress, Incorrect. These labels indicate how close a student was to the correct solution, provided by the instructor.

Upon reviewing the TA's identifications and the corresponding student work, the instructor found only 3 disagreements out of 408 instances; see Table 3. One particular disagreement involved an analysis of running time, where the instructor concluded that a student's response, although more detailed than usual, was accurate, contrary to the TA's initial assessment. The other two discrepancies concerned issues of syntax and logic errors identified by the TA.

5.3. Student Perception of The Learning Experience

An end-of-the-semester anonymous survey was conducted to gather feedback from students regarding the impact of teaching assistant involvement in lectures, particularly during the in-class coding activities. Twenty two out of 26 students participated in the anonymous survey. The overwhelming majority of the students found that the in-class coding experience was useful. Specifically, 92% of the participants answered 9 or 10 to the question: "On a 0-10 scale (0 is worse, 10 is best), how useful did you find the experience of participating in in-class exercises?". Table 4 shows students' response to four specific questions about the in-class exercise experience. We found that:

- Students were eager to participate in in-class exercises. 96% of the students regularly check their submission status to see if they were graded. Further, 96% of them reviewed feedback provided by the instructor or TA.
- Students thought the experience to be useful to them in different ways. 87% of them found that mistakes they made on the in-class exercises were reviewed and discussed in class by the instructor. 82% of them reported that they received help from the instructor or TA.
- There was room for improvement. Although 40% of students were completely satisfied with the learning experience, about 30% of them had some technical issues in terms of learning or using the platform. Further, 25% of them suggested that certain aspects of the active learning experience could be improved. Finally, 5% of them suggested to improve on the topics of the exercises.

These findings suggest that active TA involvement in lectures, through the facilitation of coding activities and provision of immediate support, enhances student engagement and learning outcomes.

Table 3

TA's assessment of student progress

Did the instructor agree with the TA's assessment?	TA's assessment	Cases
Yes	Adequate	260
	No Progress	61
	In Progress	10
	Incorrect	74
No	Adequate	2
	No Progress	0
	In Progress	0
	Incorrect	1

Table 4
End-of-semester Student Survey

	Always	Often	Sometimes	Rarely
Did you often check to see if your submissions were graded?	70%	26%	4%	0%
How often did you review feedback provided by the instructor or TA?	87%	9%	4%	0%
Did the instructor discuss mistakes in your submitted work in class?	78%	10%	4%	8%
Did you get help from the TA or instructor when you were stuck?	65%	17%	9%	9%

5.4. How TAs' Feedback Differs from ChatGPT's

Uninfluenced Feedback Evaluation

To assess the potential differences in feedback quality between ChatGPT and TAs when neither were provided with any guideline on feedback provision, we employed the Mann-Whitney U test. This non-parametric test was chosen for its suitability for analyzing ordinal data without assuming a normal distribution of the underlying data. The comparison focused on feedback scores as depicted in Fig. 4.

Our analysis revealed that TAs outperformed ChatGPT in several key areas, notably in *technical quality*, *ability to avoid revealing answers directly*, and *effectiveness in identifying learning gaps* within student submissions. For these criteria, the p-values associated with the comparison between TAs and ChatGPT were below 0.05, indicating statistical significance.

Conversely, ChatGPT demonstrated superior performance in *providing motivational support*, *structuring feedback in complete sentences*, and *incorporating examples to clarify feedback*. These areas saw ChatGPT achieving p-values less than 0.05 when compared to TAs, suggesting notable strengths in these dimensions.

However, in the realm of delivering actionable feedback that directly guides students towards resolving their learning challenges, our findings indicated no significant difference between the feedback provided by TAs and ChatGPT.

In terms of responsiveness, the TAs, TA1 and TA2, required an average of 44 seconds and 59.2 seconds, respectively, to generate feedback for each student submission. The standard deviations for their response times were 15.22 seconds for TA1 and 41.83 seconds for TA2, highlighting variability in their feedback timeliness. In stark contrast, ChatGPT demonstrated a capability to furnish feedback within a few seconds for all evaluated submissions, showcasing its efficiency in response time.

Guided Feedback Evaluation

Within this analysis, both ChatGPT and TAs received a set of guidelines aimed at enhancing the effectiveness of their feedback. Initial

observations highlighted that, despite these guidelines, TAs maintained a superior level of technical quality in their feedback compared to ChatGPT, as illustrated in Figs. 4 and 5.

The implementation of feedback guidelines led to improvements in several feedback dimensions for both ChatGPT and the TAs, albeit in varying manners, summarized in Fig. 5.

Following the guidelines, ChatGPT showed significant improvement (with p-value less than 0.05) in the levels of encouragement and in effectively concealing answers from students, demonstrating its adaptability to guideline-driven feedback enhancement. However, other feedback elements such as technical quality and example usage by ChatGPT remained consistent, showing no significant change post-guideline introduction.

TA1 exhibited a notable adherence to the guidelines, which significantly enhanced the encouragement level, the delivery of actionable feedback, and the utilization of complete sentences in their feedback, with all associated p-values being less than 0.05. This adherence not only clarified next steps for students but also promoted a fuller use of complete sentences, enhancing feedback clarity. Unlike ChatGPT, changes in other feedback aspects were minimal or nonexistent for TA1.

Conversely, TA2 showed a lack of adherence to the feedback provision guidelines. In a subsequent interview, TA2 articulated a view that many elements of the effective feedback guidelines were superfluous, potentially leading to feedback that was too verbose. This approach by TA2 did not influence the technical quality or the use of examples in their feedback, which remained consistent.

Similar to the findings from the uninfluenced feedback analysis, the time taken by TA1 and TA2 to provide feedback post-guideline introduction averaged 51 seconds and 51.97 seconds, respectively, with standard deviations of 13.98 and 20.77 seconds. Compared to the previous scenario where TAs were not guided to provide feedback, on average, TA1 took about 7 seconds longer, and TA2 took 7 seconds fewer to provide feedback.

5.5. Effectiveness of Collaboration Between TAs and ChatGPT

TA Taking Advantage of ChatGPT's Assistance

This study aimed to investigate the potential improvement in feedback quality when TAs review and augment ChatGPT's initial feedback on student code submissions. We termed this augmented feedback as *AI-empowered feedback*. The primary research question was to determine if AI-empowered feedback by a TA is superior in quality to feedback provided solely by ChatGPT or the TA's unaided feedback. For the analysis of these paired data sets, the Wilcoxon signed-rank test was utilized, which is a fitting non-parametric statistical test for such comparisons.

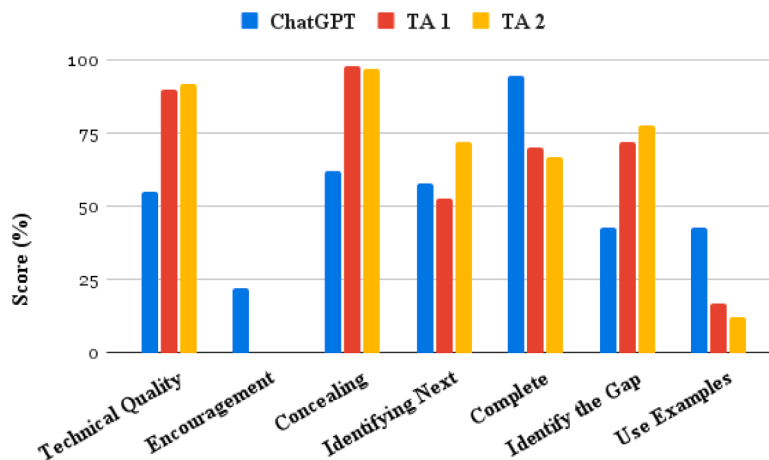


Fig. 4. Feedback scores of ChatGPT and TAs prior to having a guideline on effective feedback.

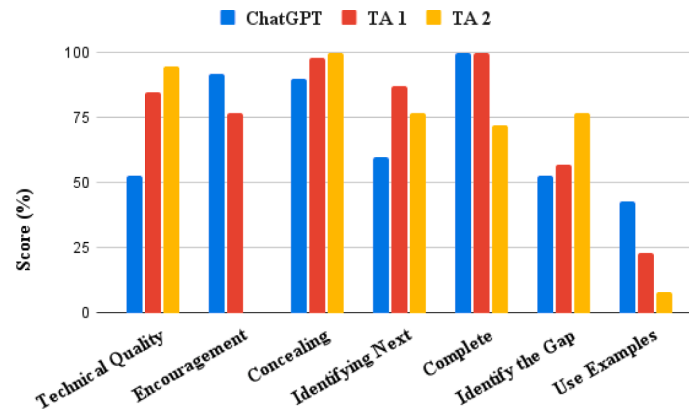


Fig. 5. Feedback scores of ChatGPT and TAs after having a guideline on effective feedback.

The experiment focused exclusively on TA1, as indicated in Fig. 6, with TA2 not participating in this portion of the study. The results showed that after the TA revised ChatGPT's feedback, the resulting feedback was improved across all evaluated criteria, notably in terms of technical quality, identifying the next action, and the identification of technical and conceptual gaps, with all p-values being less than 0.05. In terms of "technical quality", 97% of the AI-empowered TA's feedback was found to be of high quality; this is 12% improvement. In terms of, "identifying the next action", 92% was found to be of high quality; a 5% improvement. In terms of "identifying the gap", 70% was found to be of high quality; a 13% improvement.

Furthermore, a comparison was made between TA1's AI-empowered feedback and his original feedback on the same submissions. This comparison revealed significant enhancements in feedback effectiveness across the board. The improvements were particularly significant in the areas of technical quality and the identification of technical and conceptual gaps, with associated p-values all below 0.05.

On average, TA1 spent 59 seconds (with a standard deviation of 9.6 seconds) on each student submission to generate AI-empowered feedback during this experimental phase. This shows that on average the TA took 8 seconds more to provide feedback when he consulted with ChatGPT.

ChatGPT Taking Advantage of TA's Assistance

This experiment delved deeper into the concept of collaborative feedback by first having TAs review ChatGPT's feedback, and then providing such feedback to ChatGPT to further revision. This way ChatGPT can take advantage of TA strengths and improve upon them. The procedure was structured as follows: (1) ChatGPT generated initial feedback on a student's code submission; (2) TA1 or TA2 reviewed and modified this feedback based on their expertise; and (3) ChatGPT

evaluated the TA's enhancements to provide further refined feedback.

The result was shown in Fig. 7. ChatGPT was to take advantage of TA1's feedback, leading to a significant 30% improvement in terms of technical quality (p-value less than 0.05), a 27% improvement in terms of identifying the next action for students, and a 19% improvement on identifying the gap.

However, the collaboration with TA2 did not yield similarly positive results, indicating a potential discrepancy in the effectiveness of the feedback process between the TAs. This difference highlights the critical role of synergy between human expertise and AI capabilities in optimizing feedback quality.

6. Discussion

Summary of findings: We proposed an approach that utilizes in-class exercises as a form of active learning and formative assessment. We aimed to investigate the impact of TAs and the interactions between TAs and generative AI within this environment. We discovered that the use of TAs could be very effective, and additional assistance from generative AI could further improve the learning experience.

Specifically, our TA accurately identified whether students were struggling or making progress during these exercises. We found substantial agreement between the TA's assessment of student progress and a post-hoc evaluation by the course instructor. Consequently, the TA was not only able to identify struggling students but also effectively pinpointed a topic, namely *backtracking*, with which students struggled the most. An end-of-semester survey revealed strong student engagement and satisfaction with the learning experience.

Furthermore, we found that providing TAs and ChatGPT with guidelines on effective feedback actually improved their feedback. We

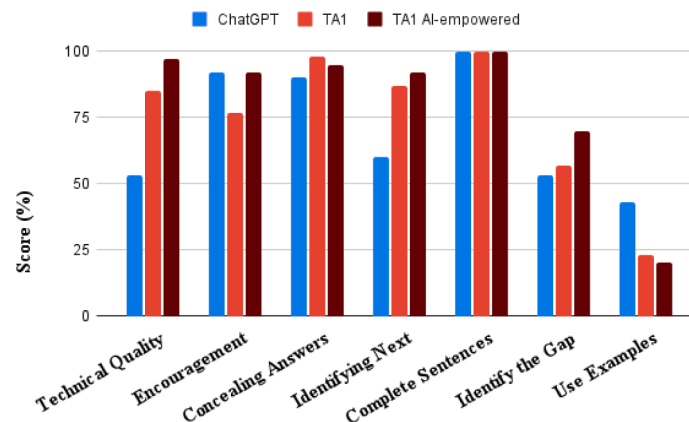


Fig. 6. Scores of TA1's original feedback, ChatGPT's feedback, and TA1's AI-empowered feedback.

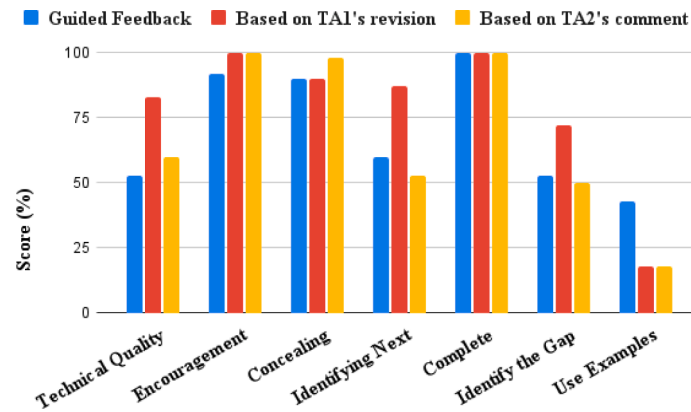


Fig. 7. Feedback scores for ChatGPT with a guideline versus ChatGPT after receiving feedback from TAs.

showed that why human TAs were better at providing technical assessment, identifying conceptual gaps and next actions, ChatGPT was better at providing effective clarifying examples and encouragements. Moreover, we found that while ChatGPT endeavored to comply with the guideline, a human TA might resist parts of it. In our case, one of the TAs refrained from providing encouragement to students, although the guideline recommended it. Most importantly, when TAs and ChatGPT revised each other's feedback to students, the quality of the feedback improved in several aspects, including technical quality and the identification of conceptual gaps, as well as suggesting the next actions for students to take.

Potential Impacts on AI Literacy: The proposed approach promises to offer a scalable and relatively inexpensive solution to enhance AI literacy. With the help of generative AI and TAs, an instructor can manage large classrooms that support numerous students remotely in real-time. Simultaneously, the learning environment promotes an active, hands-on instructional style, which is beneficial for teaching highly technical concepts. The widespread adoption of remote instruction in the post-COVID era globally [72] further underscore the potential impact of this approach. Lastly but most importantly, the collaboration between human TAs and generative AI facilitates the identification of individual learning needs and enables immediate tailored interventions. As education in computing and AI becomes increasingly widespread, this approach holds the promise of making education more accessible and inclusive.

Future Directions: While this work lays a foundation for instruction, additional studies are needed to optimize and enhance various aspects. First, there is potential for identifying designs of in-class exercises that are effective in this environment. Well-designed exercises can not only help students incrementally improve their knowledge and skills but also assist human teachers in more effectively identifying gaps in students' understanding. Second, there is potential for improving classroom logistics to make learning more efficient. Our study reported that when TAs and ChatGPT collaborated, TAs took slightly longer to provide feedback to students. Although the effectiveness of feedback increased, there is still room to enhance efficiency. Finally, further investigation is needed into how human teachers (TAs and instructors) can collaborate effectively and synergistically with generative AI. We have identified some unique strengths of both human teachers and ChatGPT, but much more work can be done in this direction. For collaboration between a human teacher and AI to be successful, there must be synergy and a mutual understanding of each other's strengths, weaknesses, and peculiarities.

Threats to Validity and Limitations: This research is limited in scope to a few courses and TAs, which may constrain the generalizability of its conclusions. The variability and potential bias introduced by the limited number of TAs further challenge the universal applicability of our findings. Given that the data originates exclusively from in-class

exercises, the results may reflect conditions unique to the selected courses and the instructional strategies employed therein. Additionally, the choice of exercises and the TAs' interpretations of the feedback guidelines could have influenced the observed outcomes.

CRediT authorship contribution statement

Kritish Pahi: Writing – review & editing, Writing – original draft, Software, Validation, Methodology, Data curation, Conceptualization. **Shiplu Hawlader:** Writing – review & editing, Writing – original draft, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **Eric Hicks:** Writing – review & editing, Data curation. **Alina Zaman:** Writing – review & editing, Methodology. **Vinhthuy Phan:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] Phan V, Hicks E. Code4brownies: an active learning solution for teaching programming and problem solving in the classroom. *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. 2018. p. 153–8.
- [2] Hicks NM, Douglas KA. Efforts to improve undergraduate grader consistency: a qualitative analysis. *2018 ASEE Annual Conference & Exposition*. 2018.
- [3] Hicks E, Cook A, Malasri K, Zaman A, Phan V. Keep it relevant! Using in-class exercises to predict weekly performance in CS1. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1*. 2022. p. 154–60.
- [4] Hicks E, Tran Q, Malasri K, Vo NS, Phan V. Active learning: the almost silver bullet. *2020 12th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE; 2020. p. 131–5.
- [5] Cook A, Phan V, Windsor A. Improving TA feedback on in-class coding assignments for introductory computer science. *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*. 2022. p. 421–7.
- [6] Cook A, Zaman A, Hicks E, Malasri K, Phan V. Try that again! How a second attempt on in-class coding problems benefits students in CS1. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1*. 2022. p. 509–15.
- [7] Kandlhofer M, Steinbauer G, Hirschmugl-Gaisch S, Huber P. Artificial intelligence and computer science in education: from kindergarten to university. *2016 IEEE frontiers in education conference (FIE)*. IEEE; 2016. p. 1–9.
- [8] Long D, Magerko B. What is AI literacy? Competencies and design considerations. *Proceedings of the 2020 CHI conference on human factors in computing systems*. 2020. p. 1–16.
- [9] Druga S, Vu ST, Likhith E, Qiu T. Inclusive AI literacy for kids around the world. *Proceedings of FabLearn 2019*. 2019. p. 104–11.
- [10] Julie H, Alyson H, Anne-Sophie C. Designing digital literacy activities: an interdisciplinary and collaborative approach. *2020 IEEE Frontiers in Education Conference (FIE)*. IEEE; 2020. p. 1–5.

- [11] Vazhayil A, Shetty R, Bhavani RR, Akshay N. Focusing on teacher education to introduce AI in schools: perspectives and illustrative findings. 2019 IEEE tenth international conference on Technology for Education (T4E). IEEE; 2019. p. 71–7.
- [12] Han X, Hu F, Xiong G, Liu X, Gong X, Niu X, Shi W, Wang X. Design of AI+ curriculum for primary and secondary schools in qingdao. 2018 Chinese automation congress (CAC). IEEE; 2018. p. 4135–40.
- [13] How M-L, Hung WLD. Educating AI-thinking in science, technology, engineering, arts, and mathematics (STEAM) education. *Educ Sci* 2019;9(3):184.
- [14] Chai CS, Lin P-Y, Jong MS-Y, Dai Y, Chiu TKF, Huang B. Factors influencing students' behavioral intention to continue artificial intelligence learning. 2020 international symposium on educational technology (ISET). IEEE; 2020. p. 147–50.
- [15] Gong X, Tang Y, Liu X, Jing S, Cui W, Liang J, Wang F-Y. K-9 artificial intelligence education in qingdao: Issues, challenges and suggestions. 2020 IEEE international Conference on networking, Sensing and control (ICNSC). IEEE; 2020. p. 1–6.
- [16] Fui-Hoon Nah F., Zheng R., Cai J., Siau K., Chen L.. Generative AI and chatGPT: applications, challenges, and AI-human collaboration. 2023.
- [17] Touretzky D, Gardner-McCune C, Martin F, Seehorn D. Envisioning AI for k-12: what should every child know about AI? Proceedings of the AAAI conference on artificial intelligence. vol. 33; 2019. p. 9795–9.
- [18] Shemshack A, Spector JM. A systematic literature review of personalized learning terms. *Smart Learn Environ* 2020;7(1):1–20.
- [19] Grant P, Basye D. Personalized learning: a guide for engaging students with technology. International Society for Technology in Education; 2014.
- [20] of Educational Technology O. Reimagining the role of technology in education: 2017 national education technology plan update. Report. Washington, D.C.: US Department of Education; 2017.
- [21] Bulger M. Personalized learning: the conversations were not having. *Data Soc* 2016;22(1):1–29.
- [22] Kalyuga S, Ayres P, Chandler P, Sweller J. The expertise reversal effect. *Educ Psychol* 2003;38(1):23–31.
- [23] Bahçeci F, Gürol M, et al. The effect of individualized instruction system on the academic achievement scores of students. *Educ Res Int* 2016;2016.
- [24] Arnesen KT, Graham CR, Short CR, Archibald D. Experiences with personalized learning in a blended teaching course for preservice teachers. *J Online Learn Res* 2019;5(3):275–310. **Special Issue: Journal Of Online Learning Research**
- [25] Deng Y, Lu D, Chung C-J, Huang D, Zeng Z. Personalized learning in a virtual hands-on lab platform for computer science education. 2018 IEEE Frontiers in education conference (FIE). IEEE; 2018. p. 1–8.
- [26] Kopeyev Z, Mubarakov A, Kultan J, Aimicheva G, Tuyakov Y. Using a personalized learning style and google classroom technology to bridge the knowledge gap on computer science. *Int J Emerg Technol Learn (IJET)* 2020;15(2):218–29.
- [27] Campbell LO, Cox TD. Digital video as a personalized learning assignment: a qualitative study of student authored video using the ICSDR model. *J Scholarship Teach Learn* 2018;18(1):11–24.
- [28] Zhang L, Basham JD, Yang S. Understanding the implementation of personalized learning: a research synthesis. *Educ Res Rev* 2020;31:100339.
- [29] Vygotsky LS, Cole M. Mind in society: development of higher psychological processes. Harvard university press; 1978.
- [30] Kim MC, Hannafin MJ. Scaffolding problem solving in technology-enhanced learning environments (TELEs): bridging research and theory with practice. *Comput Educ* 2011;56(2):403–17.
- [31] Bruner JS. The act of discovery. *Harvard Educ Rev* 1961.
- [32] Tobias J, Duffy TM. Constructivist instruction: success or failure? Routledge; 2009.
- [33] Sweller J. Cognitive load during problem solving: effects on learning. *Cognit Sci* 1988;12(2):257–85.
- [34] Crippen KJ, Earl BL. The impact of web-based worked examples and self-explanation on performance, problem solving, and self-efficacy. *Comput Educ* 2007;49(3):809–21.
- [35] Margulieux LE, Catrambone R. Improving problem solving with subgoal labels in expository text and worked examples. *Learning and Instruction*. vol. 42. Elsevier; 2016. p. 58–71.
- [36] Hadjerrouit S. A constructivist framework for integrating the java paradigm into the undergraduate curriculum. Proceedings of the 6th Annual Conference on the Teaching of Computing and the 3rd Annual Conference on Integrating Technology into Computer Science Education: Changing the Delivery of Computer Science Education. New York, NY USA: ACM; 1998. p. 105–7.
- [37] Craft A. Neuro-linguistic programming and learning theory. *Curriculum J* 2001;12(1):125–36.
- [38] Wulf T. Constructivist approaches for teaching computer programming. Proceedings of the 6th Conference on Information Technology Education (SIGITE 2005). New York, NY USA: ACM; 2005. p. 245–8.
- [39] Lui AK, Kwan R, Poon M, Cheung YHY. Saving weak programming students: applying constructivism in a first programming course. *ACM SIGCSE Bull* 2004;36(2):72–6.
- [40] Guo PJ. Codeopticon: real-time, one-to-many human tutoring for computer programming. Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology. 2015. p. 599–608.
- [41] Zhang AG, Chen Y, Oney S. Vizprog: identifying misunderstandings by visualizing students coding progress. Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. 2023. p. 1–16.
- [42] Fenske RH, Geranios CA, Keller JE, Moore DE. Early intervention programs: Opening the door to higher education. Washington, DC: ERIC Publications; 1997.
- [43] Zhang Y, Fei Q, Quddus M, Davis C. An examination of the impact of early intervention on learning outcomes of at-risk students. *Res Higher Educ J* 2014;26.
- [44] Gordanier J, Hauk W, Sankaran C. Early intervention in college classes and improved student outcomes. *Econ Educ Rev* 2019;72:23–9.
- [45] Liao SN, Zingaro D, Laurenzano MA, Griswold WG, Porter L. Lightweight, early identification of at-risk CS1 students. Proceedings of the 2016 acm conference on international computing education research. 2016. p. 123–31.
- [46] Hung J-L, Wang MC, Wang S, Abdelrasoul M, Li Y, He W. Identifying at-risk students for early interventions a time-series clustering approach. *IEEE Trans Emerg Top Comput* 2015;5(1):45–55.
- [47] Decker A, Ventura P, Egert C. Through the looking glass: reflections on using undergraduate teaching assistants in CS1. Proceedings of the 37th SIGCSE technical symposium on Computer science education. 2006. p. 46–50.
- [48] Dickson PE, Dragon T, Lee A. Using undergraduate teaching assistants in small classes. Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. 2017. p. 165–70.
- [49] O'neal C, Wright M, Cook C, Perorazio T, Purkiss J. The impact of teaching assistants on student retention in the sciences: lessons for TA training. *J College Sci Teach* 2007;36(5):24.
- [50] Roberts E, Lilly J, Rollins B. Using undergraduates as teaching assistants in introductory programming courses: an update on the stanford experience. Proceedings of the twenty-sixth SIGCSE technical symposium on Computer science education. 1995. p. 48–52.
- [51] Ellis J, Deshler J, Speer N. Supporting institutional change: a two-pronged approach related to graduate teaching assistant professional development. Proceedings of the 19th Annual Conference on Research in Undergraduate Mathematics Education. 2016. p. 1–7.
- [52] Estrada FJ, Tafilovich A. Bridging the gap between desired and actual qualifications of teaching assistants: an experience report. Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education. 2017. p. 134–9.
- [53] Gerritsen D. A socio-technical approach to feedback and instructional development for teaching assistants. PhD diss, Carnegie Mellon University 2018.
- [54] Pahi K, Phan V. A cloud-based technology for conducting in-class exercises in data science and machine learning courses. Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1. 2023. p. 868–74.
- [55] Zaman A, Cook A, Phan V, Windsor A. A practical strategy for training graduate CS teaching assistants to provide effective feedback. Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1. 2023. p. 285–91.
- [56] Lo CK. What is the impact of chatGPT on education? A rapid review of the literature. *Educ Sci* 2023;13(4):410.
- [57] Haleem A, Javaid M, Singh RP. An era of chatGPT as a significant futuristic support tool: a study on features, abilities, and challenges. *BenchCouncil Trans Benchmark Standard Eval* 2022;2(4):100089.
- [58] Baidoo-Anu D, Owusu Ansah L. Education in the era of generative artificial intelligence (AI): understanding the potential benefits of chatGPT in promoting teaching and learning. Available at SSRN 4337484 2023.
- [59] Chiu TKF, Moorhouse BL, Chai CS, Ismailov M. Teacher support and student motivation to learn with artificial intelligence (AI) based chatbot. *Interact Learn Environ* 2023;1–17.
- [60] Gill SS, Kaur R. ChatGPT: vision and challenges. *Internet of Things Cyber-Phys Syst* 2023;3:262–71.
- [61] Zhang C, Zhang C, Li C, Qiao Y, Zheng S, Dam SK, Zhang M, Kim JU, Kim ST, Choi J, et al. One small step for generative ai, one giant leap for agi: a complete survey on chatgpt in aigc era. *arXiv preprint arXiv:230406488* 2023.
- [62] Singh A, Karayev S, Gutowski K, Abbeel P. Gradescope: a fast, flexible, and fair system for scalable assessment of handwritten work. Proceedings of the fourth (2017) acm conference on learning@ scale. 2017. p. 81–8.
- [63] Tulasi L. Integration of AI-technologies into ELT: a brief study. *J Res Scholars Professional English Language Teach* 2023;7(38).
- [64] Biehler R, Fleischer Y. Introducing students to machine learning with decision trees using CODAP and jupyter notebooks. *Teach Stat* 2021;43:S133–42.
- [65] Butler DL, Winne PH. Feedback and self-regulated learning: a theoretical synthesis. *Rev Educ Res* 1995;65(3):245–81.
- [66] Yen Y-CG, Dow SP, Gerber E, Bailey BP. Listen to others, listen to yourself: combining feedback review and reflection to improve iterative design. Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition. 2017. p. 158–70.
- [67] Ertmer PA, Richardson JC, Belland B, Camin D, Connolly P, Coulthard G, Lei K, Mong C. Using peer feedback to enhance the quality of student online postings: an exploratory study. *J Comput-Mediat Commun* 2007;12(2):412–33.
- [68] Hattie J, Timperley H. The power of feedback. *Rev Educ Res* 2007;77(1):81–112.
- [69] Deci EL, Koestner R, Ryan RM. A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. *Psychol Bull* 1999;125(6): 627.
- [70] Lovett M, Meyer O, Thille C. The open learning initiative: measuring the effectiveness of the OLI statistics course in accelerating student learning. *J Interact Media Educ* 2008.
- [71] Gibbs G, Simpson C. Conditions under which assessment supports students learning. *Learn Teach Higher Educ* 2005;1(1):3–31.
- [72] Khan S, Zayed NM, Darwish S, Nitsenko V, Islam KA, Hassan MA, Dubrova O. Pre and present COVID-19 situation: a framework of educational transformation in south asia region. *Emerg Sci J* 2022;7:81–94.