

Learning by teaching with ChatGPT: The effect of teachable ChatGPT agent on programming education

Angxuan Chen¹ | Yuang Wei²  | Huixiao Le¹ | Yan Zhang³

¹Department of Educational Technology,
Peking University, Beijing, China

²Lab of Artificial Intelligence for
Education, East China Normal University,
Shanghai, China

³Department of Computer Science
and Technology, Tsinghua University,
Beijing, China

Correspondence

Yuang Wei, Lab of Artificial Intelligence for
Education, East China Normal University,
Shanghai, China.

Email: philrain@foxmail.com

Abstract: This study investigates the potential of using ChatGPT as a teachable agent to support students' learning through teaching, specifically in programming education. While learning by teaching (LBT) is an effective pedagogical strategy, traditional teachable agents often struggle with facilitating dynamic, dialogue-based interactions. Our research explored whether ChatGPT, with its advanced conversational capabilities, can effectively support this process. Findings indicate that teaching ChatGPT improved students' knowledge gains and programming abilities, particularly in writing readable and logically sound code. However, its impact on error-correction skills was limited, likely due to ChatGPT's tendency to generate correct code, thus reducing debugging opportunities. Notably, students' self-regulated learning (SRL) abilities improved, suggesting that the act of teaching ChatGPT enhances learners' self-efficacy and SRL strategy implementation. The study discusses how engaging in instructional dialogues with an artificial intelligence (AI) can contribute to the LBT process and explores ChatGPT's specific role in supporting students' SRL. Overall, this research highlights ChatGPT's potential as a teachable agent, offering insights for future research on AI-driven pedagogical tools and their broader implications for education.

KEYWORDS

ChatGPT, programming education, self-regulated learning, teachable agent

Practitioner notes

What is already known about this topic?

- Learning by teaching (LBT) is an effective instructional method that fosters learners' active learning, often within a social context.
- Previous studies have explored the effect of teachable agents as virtual students, but most agents lack support for dynamic, dialogue-based interactions in the LBT process.
- Currently, ChatGPT shows potential to facilitate flexible teaching interactions and allows learners to adapt their teaching strategies. However, the effectiveness of teachable ChatGPT agents remains largely unknown.

What this paper adds?

- This study showed that learning with a teachable ChatGPT agent improved learners' knowledge gains, programming abilities and SRL abilities.
- This study also found that teaching a teachable ChatGPT agent did not significantly benefit learners' error-correction abilities.

Implications for practice and/or policy

- Educators can encourage learners to teach or explain concepts to ChatGPT agents through dialogue to promote deeper cognitive processes.
- When designing ChatGPT-based teachable agents in programming education, errors should be carefully designed to ensure that students engage with meaningful and representative programming challenges.
- Future research should explore how SRL strategies can be better integrated into the interactions between students and ChatGPT-based teachable agents.

INTRODUCTION

Learning by teaching (LBT) is an innovative instructional method that has garnered significant attention in educational research in recent years. LBT requires learners not only to express and reorganize existing knowledge but also to actively engage in reflective knowledge construction. This approach to knowledge construction goes beyond simple knowledge transmission, encouraging learners to create and refine knowledge that is useful for learning by forming deeper explanations, analogies and connections across broader contexts (Chase et al., 2009; Chi & Wylie, 2014). Most studies investigating LBT entailed interacting with an audience (Duran, 2017; Roscoe & Chi, 2007). Considering the limited availability of peers for teaching exercises, past research has looked into utilizing artificial intelligence (AI) as 'teachable agents' (TAs) to support in the LBT method. These agents are programmed to pose questions and make demands, encouraging learners to teach them. When used as an instructional activity, students are often explicitly told to adopt a teaching role by communicating distinct information on the subject matter to a fictitious fellow student with low levels of prior knowledge (Lachner et al., 2019). Through this interactive teaching process, learners not only instruct the agents but also deepen their own understanding and mastery of the subject matter (Silvervarg et al., 2021). Empirical results supported the effectiveness of TAs on achieving deeper learning and transfer (Kim & Baylor, 2016), emotional regulation (Han et al., 2021) and motivation (Kim & Lim, 2013).

However, the development and implementation of TAs are not without challenges. One of the primary limitations is the technical capability of these agents. Most TAs developed thus far can only handle predefined problems and are limited in their interaction capabilities, often restricted to non-verbal responses or simplistic interactions, such as button clicking or filling in blanks (eg, Biswas et al., 2016; Pareto et al., 2009). This limitation stems from current gaps in technology that limit these agents from processing and responding to natural language inputs in a way that mimics human-like interactions. As a result, these agents are often unable to initiate or sustain interactions that are perceived as socially rich and engaging by human standards. As previous studies pointed out (Lachner et al., 2022), LBT is inherently social. The limited interaction capabilities of these agents hinder learners' engagement and reflection. For instance, research has shown that the effectiveness of LBT varies depending on the user's belief about whether they are interacting with a human or a computer agent. Even when the agents' responses are identical, students tend to offer less detailed self-explanations and respond with brief keywords when interacting with an agent, compared with when they believe they are interacting with a human (Ogan et al., 2012).

Currently, the rapid development of generative AI, particularly models like ChatGPT, has introduced a natural conversation method that is capable of engaging in much deeper and more sophisticated conversations with learners (Liu et al., 2023; Lo, 2023). Unlike traditional TAs that are limited by predefined problem sets and simplistic interaction modes, ChatGPT and similar generative AI models can process natural language inputs and generate human-like responses (Fui-Hoon Nah et al., 2023). Research has shown that learners often perceive ChatGPT as a human-like agent, especially when it assists with learning tasks (Luo, 2024; Markel et al., 2023). However, there is limited research on the effects of using ChatGPT as a teachable agent and the teaching strategies employed in such environments. This study investigated the impact of ChatGPT-based TAs in a programming learning task and examined learners' knowledge gains, programming skills and self-regulated learning (SRL) abilities. The goal was to explore ChatGPT's potential in enhancing the LBT process.

BACKGROUND AND RELATED WORKS

Learning by teaching

As a widely recognized instructional method, LBT not only encourages learners to express and reorganize their existing knowledge but also actively engages them in reflective knowledge construction. Unlike simple knowledge transmission, LBT promotes the creation and refinement of understanding by guiding learners to develop deeper explanations, analogies and connections across diverse contexts (Chase et al., 2009; Chi & Wylie, 2014; Duran, 2017; Roscoe & Chi, 2007). At the core of LBT is the idea that teaching prompts learners to move beyond merely expressing what they already know. Through reflection and reorganization, they develop a deeper understanding and construct a new knowledge system. This knowledge system in LBT is crucial; it not only draws on learners' existing knowledge but also challenges them to extend this knowledge beyond the provided materials, building more complex and nuanced cognitive structures (Nasir & Lee, 2014). On the contrary, teaching itself is a highly reflective process. Learners must continually monitor and adjust their understanding to ensure that they can clearly convey knowledge to others. This reflective process naturally enhances learners' metacognitive abilities, making them more conscious and proactive in monitoring their cognitive processes and learning progress (Kinnebrew et al., 2013; Spruce & Bol, 2015). In the teaching process, learners become aware of their own knowledge gaps through teaching others. These 'knowledge gaps' prompt them to take action and further refine their understanding, thereby strengthening their self-regulation (Biswas, Leelawong,

Belynne, Viswanath, Schwartz, et al., 2004). Lastly, LBT fosters a sense of responsibility and motivation in students, encouraging them to take greater ownership of planning and managing their learning tasks, which further promotes the development of both cognitive and metacognitive processes in self-regulation (Biswas et al., 2009; Biswas, Leelawong, Belynne, Viswanath, Vye, et al., 2004). The processes of self-reflection, knowledge restructuring, self-monitoring and self-regulation that occur during learning fall under the scope of SRL (Puustinen & Pulkkinen, 2001; Zimmerman, 2002) and combining the SRL model with the LBT approach has been proven to be a more effective theoretical way to enhance learning outcomes (Biswas et al., 2005).

Despite the theoretical potential of LBT, its practical application in classrooms faces notable challenges. First, LBT demands students to alternate between the roles of teacher and learner, which is time-consuming and may hinder its widespread adoption in classroom settings (King et al., 1998). Second, studies suggest that while the tutor's learning is enhanced, the learner might not benefit as much, potentially creating an imbalance in learning outcomes (King et al., 1998; Walker et al., 2014). In response to these challenges, researchers have explored technological solutions, with TAs emerging as a prominent approach to overcoming LBT's limitations. TAs are virtual agents designed to learn from students' explanations and demonstrations, effectively acting as learners. By doing so, they provide scalability to LBT, as they are available at any time and help reduce psychological barriers like the fear of making mistakes or the pressure of immediate responses during teaching (Chase et al., 2009; Debbané et al., 2023). TAs are constructed around key components, such as knowledge representation, learning mechanisms, interactive interfaces, feedback systems, metacognitive abilities and social interaction. These elements enable TAs to engage students in reflective learning processes, enhancing their academic performance, self-efficacy and metacognitive skills (Chin et al., 2010).

Teachable agents in learning by teaching

More specifically, the positive impact of TAs on learning gains can be attributed to the fact that the core component of LBT is the peer learner (Debbané et al., 2023; Matsuda, Cohen, Koedinger, et al., 2010), and technology-supported TAs eliminate the need for human peers, which are typically required in traditional LBT approaches (Matsuda et al., 2012). TAs learn declarative and procedural knowledge from students' explanations and demonstrations, effectively acting as peer learners in LBT (Blair et al., 2007). For example, Jun et al. used Linear Kid as a peer in tutoring sessions and found that while varying levels of prior knowledge influenced learning gains, the problem-solving process itself did not differ significantly across conditions (Jun, 2003). Similarly, Mioduser et al. used the DynaLearn platform and showed that short-term interventions through graphical manipulation had minimal impact on learning outcomes (Bredeweg et al., 2009), whereas more interactive methods, such as games (Pareto, 2014) or 3D virtual environments (Zhao et al., 2012), significantly enhanced conceptual knowledge acquisition. Although these studies have achieved good results in their respective goals, they still share some common issues. For instance, the capabilities of TAs are limited, as they primarily provide support at the cognitive level and struggle to enhance metacognitive skills (Bredeweg et al., 2009). Additionally, the interaction methods have significant limitations; even when using games and virtual environments, the high implementation cost remains a major concern (Pareto, 2014; Zhao et al., 2012).

To address these issues, researchers have also conducted in-depth studies on strategies and effectiveness of TAs in enhancing metacognitive skills. In an experiment with Betty's Brain (Biswas et al., 2005), Biswas et al. (2010) explored the impact of TAs on metacognitive development. The study compared three interventions—LBT, SRL and an Intelligent

Coaching System (ICS)—with controlled meta-tutor feedback. Results showed that while SRL and LBT students performed equally well on tests, SRL students created more accurate concept maps, and both outperformed ICS in test scores and map accuracy. Notably, SRL students, who received metacognitive strategy feedback, exhibited more focused monitoring behaviours than LBT students, highlighting the value of cognitive and metacognitive feedback in improving learning outcomes (Biswas et al., 2010; Roscoe et al., 2013). Similarly, in Matsuda's studies with SimStudent (Matsuda, Keiser, Raizada, et al., 2010), the importance of self-explanation was examined. SimStudent solicits self-explanations by asking students to justify their feedback during tutoring sessions. Across two school studies, Matsuda found a strong correlation between the depth of self-explanation and student learning gains (Matsuda et al., 2013). Additionally, in the aforementioned studies, whether using Betty's Brain or SimStudent, it is essential to consider the possible decision paths at each step and predefine fixed templates. Executing a specific task typically requires about 1200 to 2300 lines of Java code, and developing different topics demands a significant amount of time and effort (Matsuda, 2022).

These studies, when examined alongside the theoretical framework of SRL, reveal a significant relationship between TAs and SRL principles. For example, SRL models emphasize metacognitive monitoring, where learners are encouraged to plan, monitor and reflect on their learning processes to enhance understanding and performance (Zimmerman, 2000). TAs align with this principle by requiring learners to teach virtual students, prompting them to engage in metacognitive reflection and strategy adjustment during the teaching process (Schwartz et al., 2009). Furthermore, SRL theory identifies self-efficacy as a critical factor influencing learners' persistence and effort (Pintrich, 2000). By simulating instructional tasks, TAs can foster learners' confidence and a sense of accomplishment during the teaching process (Schwartz et al., 2009). Additionally, SRL emphasizes the importance of emotional regulation in learning (Pintrich, 2000). TAs can create low-risk learning environments where learners practice managing test anxiety and emotional challenges, ultimately improving their emotional regulation skills (Baylor & Kim, 2005; Kim & Baylor, 2016).

Therefore, tools like Betty's Brain and SimStudent indeed offer a promising research direction with their interactive TA approaches, as well as methods to enhance metacognition through techniques like feedback on questions (Matsuda et al., 2013). However, their high development costs still limit their broader application. Additionally, the pre-designed templates and decision paths may fall short in accommodating unexpected real-life scenarios, leading to interactions and feedback that resemble mechanical answer selection, lacking natural communication, which is not conducive to the development of metacognitive skills (Terrace, 2005).

Reflecting on past studies, it is clear that current research on TAs has some limitations and gaps. Earlier work on metacognition did not go deep enough and missed exploring how TAs affect more complex aspects (Blair et al., 2007). Also, past research aimed at improving metacognition was too strict and lacked a natural, personalized approach (Biswas et al., 2010; Roscoe et al., 2013). The structured interactions and analysis did not do much to truly boost metacognition. Plus, creating these types of interactions takes a lot of time, is hard to adapt to new contexts and does not work well in different situations. Overcoming these challenges is still a difficult task.

Exploring the integration of LLMs in the learning by teaching approach

Fortunately, the emergence of large language models (LLMs) seems to make it possible to develop smarter, more personalized TAs that can engage in more natural interactions. LLMs

excel in contextual conversations (Ouyang et al., 2022; Ross et al., 2023), role-playing (Kong et al., 2023; Markel et al., 2023) and learning from demonstrations (Brown, 2020; Radford et al., 2019), enabling more natural and credible tutoring interactions, such as writing and explaining code on demand. Unlike traditional LBT systems that rely on limited, predefined interactions (Biswas et al., 2001; Leelawong & Biswas, 2008; Matsuda, Keiser, Raizada, et al., 2010; Pareto et al., 2011), LLM-powered agents allow flexible interactions, letting learners ask open-ended questions and explore various teaching methods, which enhances knowledge construction and metacognition (Aflalo, 2021; Chin & Brown, 2002). For instance, Markel et al.'s GPTEach project demonstrated how LLMs can simulate interactions between teaching interns and virtual students by setting roles and contexts within prompts (Markel et al., 2023). This setup provided a more realistic training environment for interns, allowing them to interact with different types of virtual students. Such agents not only facilitate flexible teaching interactions but also allow learners to adapt their teaching strategies and improve based on feedback, making the process more dynamic and productive (Biswas et al., 2001; Leelawong & Biswas, 2008; Matsuda et al., 2018).

The emergence of LLMs provides a powerful tool for creating new types of TAs. Their advanced language comprehension, flexible conversation capabilities and strong imitation skills enable them to simulate natural learning scenarios and adapt teaching strategies. These qualities make LLMs well-suited for the LBT approach. While research has shown the flexibility of LLMs in building TAs, studies on their actual impact on learning outcomes remain limited. Therefore, current research should focus on assessing the true effectiveness of LLM-based TAs and identifying areas where they can enhance learning. This calls for a deep exploration in a specific learning domain. Programming education, which involves the combined processes of knowledge acquisition and hands-on problem solving, is an ideal area for such exploration. This domain allows for better diagnosis of various learner states, such as knowledge mastery, practical skills and metacognitive attributes. Notably, in programming education, LLMs possess a broader and deeper understanding compared with typical learners. Their large-scale training across various programming languages, algorithms and best practices enables them to act as human-like programming mentors (Kazemitabaar et al., 2024). For instance, Kazemitabaar's study demonstrated how LLMs can assist students by providing instant feedback and step-by-step pseudocode explanations, offering enhanced support throughout the coding process (Kazemitabaar et al., 2023). Unlike traditional rule-based systems, LLMs offer greater flexibility, allowing students to explore multiple strategies and teaching methods without being confined to rigid instructional paths (Chen et al., 2023; Pareto et al., 2011). Moreover, LLM-powered TAs can have their cognitive behaviours, such as knowledge levels and questioning, precisely controlled (Packer et al., 2023; Zhou et al., 2023), enhancing their reasoning abilities (Chen et al., 2023; Kim et al., 2024; Lin et al., 2024) and enabling a more tailored learning experience (Junprung, 2023; Park et al., 2023). Therefore, considering the excellent capabilities of LLMs in programming education and the dual focus on measuring both knowledge and skills, we chose to delve deeper into the topic of programming education using the highly popular ChatGPT as our LLM tool.

This study will provide an in-depth analysis of ChatGPT-based TAs' impact on programming learning, focusing on the following research questions:

- RQ1: What impact does the Teachable ChatGPT Agent have on learners' knowledge gains?
- RQ2: What impact does the Teachable ChatGPT Agent have on learners' programming skills?
- RQ3: What impact does the Teachable ChatGPT Agent have on learners' SRL abilities?

METHOD

Participants

The study was conducted online during July–August 2024. This study was approved by the university's institutional review board under approval number: IRB00001052-23149. To ensure participants possessed basic programming capabilities, we specifically recruited students from the Department of Computer Science and Technology in the university. The recruitment post was conducted through the departments' official online platform, with 41 students voluntarily signing up for the experiment. All participants were between the ages of 18 and 27, with an overall average age of 21.2 (SD=2.4). The informed consent form, which was signed by all participants, stated that the data collected would be used solely for academic purposes and would not be disclosed to the public. Regarding group assignment, participants who registered in July were assigned to the control group (CG) ($N=21$), while those who registered in August were assigned to the experimental group (EG) ($N=20$). While this allocation was not strictly randomized, this approach was chosen to prevent potential information sharing between groups.

Learning materials and intervention design

The topic of our experimental learning material revolves around the 'N queens' puzzle of AI algorithm learning, which includes algorithm knowledge learning and C++ programming learning. The 'N queens' puzzle is a classic puzzle in computer science and mathematics. The challenge is to place N queens on an $N \times N$ chessboard such that no two queens can attack each other. This means no two queens can be in the same row, column or diagonal. A sample solution of 'N queens' puzzle can be found in [Figure 1](#). Here's a breakdown of the problem:

Chessboard configuration: An $N \times N$ grid, like a standard chessboard.

Queen's movement: A queen can move any number of squares vertically, horizontally or diagonally.

Objective: Place N queens on the board in such a way that no two queens can attack each other.

The participants need to calculate all possible placement coordinates of queens via programming. This problem requires students to not only understand the solution but also break it down into teachable components. Students need to:

- Explain the backtracking algorithm's working principles
- Guide ChatGPT in implementing checking functions
- Teach how to construct the recursive structure

This process simulates real teaching scenarios where students must act as instructors to guide ChatGPT's understanding and implementation. To assist participants in learning how to solve this problem through programming, we developed three online learning videos: (1) Fundamental knowledge learning Video: This video teaches participants the fundamental concepts of the backtracking algorithm used to solve this puzzle. (2) N Queen Puzzle Explanation Video: This video provides a detailed explanation of the 'N Queen' puzzle, outlining its rules and objectives. (3) Programming Solution Guide Video: In this video, a lecturer guided how to solve the puzzle with recursive concept knowledge and programming, including a step-by-step code analysis with the backtracking algorithm.

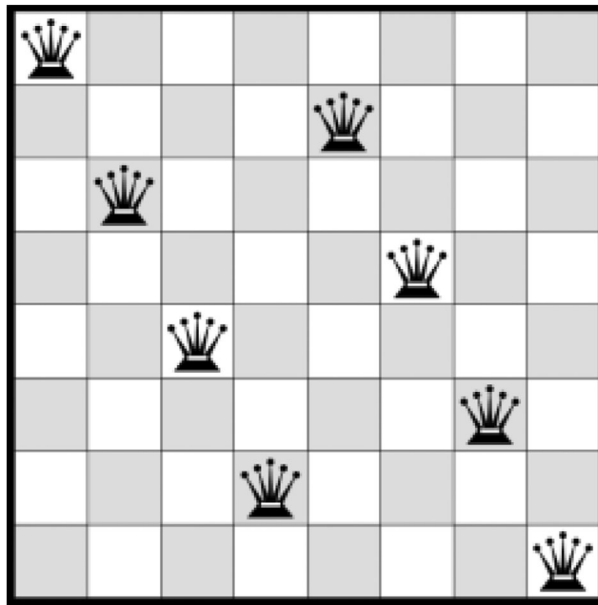


FIGURE 1 Sample solution of 'eight queens' puzzle that has no two queens in the same row, column or diagonal.

After watching these three videos, participants in the EG learned to solve the 'N Queens' puzzle with a teachable agent we designed. We employed a prompt-based design approach that could modify the role of ChatGPT. To better design ChatGPT to act as a real student seeking help from teachers (participants), we adopted the theoretical help-seeking process model proposed by Nelson-Le Gall (1981). This model describes five stages of the help-seeking process that students theoretically go through: (1) awareness of the need for help; (2) decision to seek help; (3) identification of a potential help source; (4) employment of strategies to elicit help; (5) reactions to help-seeking attempts. Based on these stages, we structured ChatGPT's responses to align with each stage by prompts, making ChatGPT a more realistic help-seeker. Specifically, we used the better version of ChatGPT, that is, gpt-4, for our experiment to ensure the prompt can be effectively understood by ChatGPT. The prompt used for this purpose can be found in the [Appendix](#). In addition, as Fui-Hoon Nah et al. (2023) mentioned, current ChatGPT may face the problem of hallucination, which means ChatGPT may generate some wrong content without control. However, when the ChatGPT agent acts as a teachable agent, those mistakes can be more like a beginner student acting and provide more opportunities for participants to teach with. Therefore, in our design, we did not chase the accuracy of the ChatGPT agent in generating correct codes but used the original ChatGPT without other augmented techniques. An example of learners' teaching the ChatGPT teachable agent can be found in [Figure 2](#).

Participants in EG were required to help the teachable ChatGPT to form correct codes that can solve the puzzle with previous instructional videos. Students were informed to explain the problem's nature, the principles of backtracking algorithms, and how to implement these concepts in code. Moreover, we developed an online judging platform that can automatically evaluate whether the participants' solutions of codes for the 'N Queens' puzzle are correct. The learning experiment was considered finished when participants successfully solved the puzzle with ChatGPT-generated codes on the online platform or hit the time limit. Additionally, participants in the CG learned to solve the 'N Queens' puzzle only through



hello, mate! let's start the eight queens puzzle learning!



Hello! I'm excited to start learning the eight queens puzzle.

1. First, I understand that the eight queens puzzle involves placing eight queens on an 8x8 chessboard in such a way that no two queens can attack each other. This means no two queens can share the same row, column, or diagonal. Is this correct?
2. I don't fully understand where to start or how to approach this problem. Could you please help me?
3. I'm choosing to ask for your help specifically because I know you have recently studied and learned how to solve the eight queens problem.
4. Could you please guide me through the initial steps? For instance, how do we begin to place the queens on the board? Should I use a specific method initially?
5. Thank you so much! I'm ready to take notes and generate the first part of the code based on your instruction.



First, you should define a bool type array to represent the chessboard. Could you show me your code of your definition?



Thank you for the guidance! If I understand correctly, we need to define a boolean type array to represent the chessboard.

From the instructions, the chessboard is 8x8, so we will need an 8x8 array. Here's what I'm thinking:

```
#include <iostream>
#include <vector>

// Define the chessboard as an 8x8 bool array
bool chessboard[8][8];
```

The `bool` type array `chessboard` can be used to represent the positions of the queens. Each cell in the array will have a value of `true` if a queen is placed in that position, and `false` otherwise.

Is this the right approach? What would be the next step in this process? Should we initialize the chessboard or move on to another part of the problem?

Thank you for your help!



正在和 聊天机器人 对话



|

发送信息

FIGURE 2 Example of learners' teaching in solving the 'N queens' puzzle.

those instructional videos, and they needed to form the solution codes by themselves to pass the examination on the online judging platform.

Procedure and data collection

All the data for this study were collected in the online setting. The complete experimental procedure is shown in Figure 3. First, all the students were required to finish a pretest online questionnaire, including a knowledge test and an SRL test. All the tests were collected online by an online questionnaire platform (<https://www.wjx.cn/>). Subsequently, the

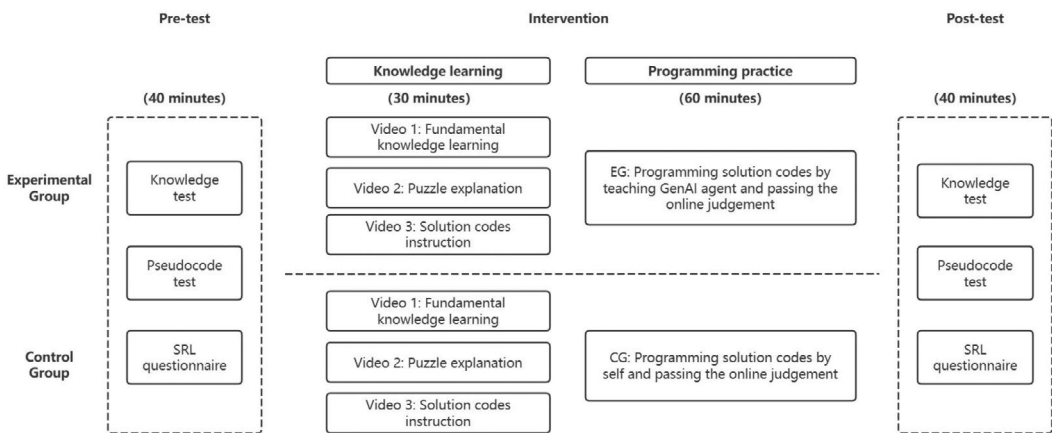


FIGURE 3 Procedure of the experiment.

researchers introduced the experiment content to all participants, including introducing the three online videos and introducing the online judging platform for their codes' submission. Second, all the participants were required to watch those three online videos to learn how to solve 'N queen' puzzle in 30 min. Third, participants in EG were trying to form the solution codes with the ChatGPT teachable agent, while participants in CG were trying to form the solution codes by themselves in the local code editor. At this stage, participants were required to share their screens via online meeting software, with one of the authors supervising to ensure they followed the video instructions and did not engage in any cheating, such as searching for solution codes online. Participants had 1 h to form correct codes by passing the online judging platform. Finally, all the participants were required to finish a posttest with a knowledge test and an SRL test. Additionally, the conversation data between the learners and the ChatGPT teachable agent were also recorded in logs for qualitative analysis. All the conversations can be found in the [Appendix](#).

Measurements

To ensure reliable measurement of knowledge test, pseudocode test and SRL ability test, we implemented specific procedures for all test and survey administrations. The same items were used in both pre and posttests to maintain direct comparability. However, to minimize potential test-reactivity effects (Song & Ward, 2015), several measures were taken: (1) participants did not receive feedback about their responses after the pre-assessments, (2) the presentation order of questions was randomized between pre and posttest for all the tests and questionnaires.

Knowledge test

The knowledge test consists of 15 single choice questions, focused on fundamental concepts of the backtracking algorithm and knowledge of 'N queens' puzzle solution (eg, What is the time complexity of the backtracking search algorithm for the 'N queens' puzzle?). The test includes five easy, five medium and five hard questions, with each question worth 1 point. The test's validity has been established through multiple validation methods. Content validity was ensured through expert review by two computer science professors

who independently evaluated the relevance and appropriateness of each question. The test also demonstrated strong construct validity with consistent internal reliability (Cronbach's $\alpha = 0.82$) in our experiment data.

Pseudocode test

Pseudocode is an algorithm description language. The purpose of using pseudocode is to enable the described algorithm to be easily implemented in any programming language with a clear structure, simple code, good readability and be similar to natural language. The well written pseudocode can represent participants' programming level (Andrzejewska et al., 2016). Our pseudocode test required participants to write the 'N queens' puzzle solution with pseudocode. The pseudocodes were evaluated by two experienced scorers, focusing on three key aspects: clearness, correctness and readability (Börstler, Bin Ali, et al., 2023) (See detailed standard in Table 1). To ensure consistency in scoring, Spearman's correlation analysis was performed, revealing high-rate agreement across all three dimensions, with correlation coefficients of 0.833, 0.935 and 0.911, all statistically significant ($p < 0.001$). Following the scoring process, the scorers engaged in cross-checking and discussions to resolve any discrepancies, ultimately agreeing on the final evaluation scores.

SRL ability test

The SRL ability was measured using an adapted version of the SRL questionnaire, which comprised 20 items rated on a five-point Likert scale. This questionnaire is derived from the MSLQ (Pintrich & De Groot, 1990), a widely recognized SRL measurement grounded in social cognitive theory (Liao et al., 2024). The MSLQ includes two dimensions, namely motivation and strategy. The motivation section focuses on learners' values, expectations and influences and is typically divided into three sub-scales: self-efficacy, intrinsic value and test anxiety. The strategy section of the adapted MSLQ emphasizes cognitive, metacognitive, and resource management strategies and is commonly divided into the Cognitive Strategies

TABLE 1 Pseudocode test scoring standard.

Dimension	Description	Range
Clearness	Clearness involves the logical flow and structure of the pseudocode, ensuring that the steps of the algorithm are straightforward and unambiguous	1–5
Correctness	Correctness is determined by the ability of the pseudocode to solve the 'N queens' puzzle as intended, ensuring that all conditions and rules of the puzzle are met without errors	1–5
Readability	Refers to how easily the pseudocode can be read and interpreted, particularly in terms of formatting, consistency and use of natural language-like constructs. Readability involves using clear variable names, consistent indentation, and comments or explanations that enhance understanding	1–5

Note: All items were rated on a 5-point Likert scale (1 = strongly disagree, 2 = disagree, 3 = neutral, 4 = agree, 5 = strongly agree). For Clearness, scorers evaluated whether the pseudocode demonstrates clear logical flow and structure. For Correctness, scorers assessed whether the pseudocode accurately solves the 'N queens' puzzle and meets all puzzle requirements. For readability, scorers evaluated whether the pseudocode is well-formatted with clear variable names and proper documentation.

Scale and the Self-Regulation Scale. Originally, the MSLQ consisted of 44 questions, but to minimize participant fatigue and ensure efficiency in data collection, it was condensed to 20 questions, specifically focusing on test anxiety, self-efficacy and cognitive strategies, as these dimensions capture key aspects of SRL.

Self-efficacy reflects learners' confidence in their ability to complete specific learning tasks. In SRL theory, it is closely linked to motivation and can significantly influence goal-setting, strategy use and persistence in learning tasks. Pintrich et al.'s SRL model (Pintrich et al., 1993) highlights the critical role of self-efficacy during the forethought phase, where it directly impacts motivational levels and effort investment. Learners with high self-efficacy are more likely to engage in metacognitive strategies, such as planning and self-monitoring, while those with low self-efficacy may avoid tasks or demonstrate minimal self-monitoring.

Cognitive strategies refer to the specific methods learners use to process and store information, such as rehearsal, elaboration and organization. These strategies are emphasized in both Pintrich et al.'s (1993) and Zimmerman's (2000) SRL models as core components of the performance phase, where learners execute their learning plans. Effective use of cognitive strategies directly impacts learning outcomes, as they reflect how actively learners engage with and process information to enhance understanding and retention.

Test anxiety represents an emotional reaction triggered by performance evaluations or test-taking scenarios and is closely tied to emotional regulation within the SRL framework, as it influences both motivation and emotional control (Panadero, 2017). According to Boekaerts et al. (2003), high test anxiety can disrupt cognitive processes by diverting cognitive resources, thus impairing learning performance. It may also lead to negative self-attributions during the self-reflection phase, further reducing future learning motivation. Conversely, low test anxiety tends to support emotional stability, fostering better execution of metacognitive strategies like self-monitoring and reflection.

Some items were modified to align with the specific focus of this study. The final Cronbach's alpha for the measurement of test anxiety, self-efficacy and cognitive strategies was 0.813, 0.791 and 0.912, respectively, indicating high reliability of the measurement used in this study.

RESULTS

Knowledge test scores

We initially analysed the pretest scores, which showed a significant difference between the EG and CG ($t=2.37$, $p=0.023$). In this situation, the knowledge test scores between EG and CG were examined using an analysis of covariance (ANCOVA), with pretest scores as the covariate and posttest scores as the dependent variable. This approach can control for pretest differences and can effectively assess knowledge gains (Borm et al., 2007). Before the analysis, the basic assumptions of the data were examined first. The test for homogeneity of regression for the dependent variable in the EG and the CG depicted no significant difference with ($F=6.27$, $p>0.05$). Similarly, the Kolmogorov–Smirnov test revealed data distributions adhering to normality ($p>0.05$). As a result, it was suitable to perform the ANCOVA test.

The ANCOVA results showed that the two groups were significantly different in knowledge test scores ($F=35.54$, $p<0.05$, $\eta^2=0.74$) (as shown in Table 2) that EG had a higher adjusted average score than CG (11.86 > 10.53). In other words, learners who learning programming knowledge with teaching the ChatGPT teachable agent have better knowledge gains than those who learning algorithm knowledge with online videos.

TABLE 2 ANCOVA results of knowledge test scores.

Group	N	Mean	SD	Adjusted mean	Adjusted SE	F	η^2
EG	20	11.90	2.09	11.86	0.32	35.54***	0.74
CG	21	10.76	1.02	10.53	0.24		

*** $p < 0.001$.

TABLE 3 ANCOVA results of programming test scores.

Dimension	Group	N	Mean	SD	Adjusted mean	Adjusted SE	F	η^2
Clearness	EG	20	4.20	0.61	4.13	0.05	7.39***	0.37
	CG	21	3.29	0.90	3.17	0.01		
Correctness	EG	20	3.81	0.96	3.79	0.10	2.98	0.19
	CG	21	3.90	1.47	3.82	0.03		
Clearness	EG	20	2.91	1.23	2.95	0.03	4.32**	0.26
	CG	21	2.81	0.58	2.83	0.02		

*** $p < 0.001$;

** $p < 0.01$.

Programming test scores

Before examining the impact of the ChatGPT teachable agent approach on participants' programming skills, Levene's test for equality of error variances showed comparable variance of the two groups ($F = 2.93$, $p > 0.05$), confirming that the variances were homogeneous. Additionally, the homogeneity of regression coefficients within the group was confirmed by the intra-group regression coefficient homogeneity test ($F = 0.13$, $p > 0.05$), allowing for the use of ANCOVA. The ANCOVA results (see Table 3) revealed that participants in the EG, who learned programming with the teachable agent, scored significantly higher in code clearness and readability compared with the CG, which learned through online videos ($F = 7.39$, $p < 0.01$, $\eta^2 = 0.37$). However, there was no significant difference in code correctness between the two groups ($F = 2.98$, $p > 0.05$, $\eta^2 = 0.19$).

SRL abilities

Before examining the impact of the ChatGPT teachable agent approach on participants' SRL abilities, Levene's test for equality of error variances was conducted and revealed no significant difference in the pretest scores between the two groups ($F = 2.07$, $p > 0.05$), indicating that the variances were homogeneous. Additionally, the test for the homogeneity of intra-group regression coefficients confirmed that the regression coefficients were consistent within each group ($F = 2.753$, $p > 0.05$), allowing for the use of ANCOVA. The ANCOVA results (See Table 4) demonstrated that the EG, which utilized the ChatGPT teachable agent, achieved significantly higher scores in self-efficacy ($F = 37.26$, $p < 0.001$, $\eta^2 = 0.75$), and cognitive strategies ($F = 18.97$, $p < 0.001$, $\eta^2 = 0.61$) compared with the CG. This suggests that learning with the ChatGPT teachable agent positively influences learners' self-efficacy and adoption of self-regulation strategies.

TABLE 4 ANCOVA results of SRL abilities.

Dimension	Group	N	Mean	SD	Adjusted mean	Adjusted SE	F	η^2
Test anxiety	EG	20	3.30	0.82	3.30	0.03	1.78	0.02
	CG	21	3.46	0.74	3.44	0.09		
Self-efficacy	EG	20	3.88	0.59	3.81	0.13	37.26***	0.75
	CG	21	3.54	0.82	3.50	0.14		
Cognitive strategies	EG	20	4.20	0.55	4.13	0.073	18.97***	0.61
	CG	21	3.76	0.62	3.77	0.11		

*** $p < 0.001$.

DISCUSSION

The effect of ChatGPT teachable agent on knowledge gains

To address RQ1, we analysed knowledge test scores and found that learners in the EG, who taught a ChatGPT teachable agent, showed significantly greater knowledge gains compared with the CG, who learned through online videos. This finding can be explained by the interactive nature of the learning process in the experimental condition. Through continuous conversation with ChatGPT, learners engaged in an active process of explaining, refining and revising their understanding. When solving the ‘N queens’ puzzle, learners needed to not only understand the puzzle’s rules and the backtracking algorithm but also respond to ChatGPT’s questions and requests for clarification. This ongoing interactive engagement required learners to critically examine their knowledge and address any gaps in their understanding, leading to deeper processing of the material compared with passive video learning (Lachner et al., 2022).

Our findings contribute to a deeper understanding of whether learners can effectively teach a teachable agent using natural language. Due to technological limitations, previous studies have rarely investigated TAs that allow learners to teach using natural language as if they were interacting with real peers (Lachner et al., 2022). In most studies involving TAs, students are usually limited to specific tasks, such as completing graphical representations or answering multiple-choice questions, rather than engaging in natural discourse (eg, Biswas et al., 2016; Chin et al., 2010; Silvervarg et al., 2021). This is significant because, as previous research has highlighted, LBT is fundamentally a social act, with its effectiveness being largely contingent on the degree of interactivity involved (Kobayashi, 2019). Our findings demonstrated that when learners engage in teaching through natural language to a ChatGPT agent, they can also achieve significant knowledge gains. This underscores the promising potential of natural language interactions in enhancing the overall learning experience with TAs.

The effect of ChatGPT teachable agent on programming skills

Our results revealed that learners in the EG achieved higher scores in programming tests related to code clearness and readability, while there were no significant differences in code correctness compared with the CG. In our learning task, code clearness refers to whether learners provided a logical flow in their coding, and readability pertains to the ease with which the code could be understood, including the use of natural language comments. These qualities are recognized as markers of high-quality code (Börstler, Bennin, et al., 2023). Our

findings suggest that by guiding a ChatGPT teachable agent in code generation, learners were able to refine their coding logic and develop a deeper understanding of what constitutes easily readable code from the perspective of the reader, rather than just the programmer. For example, a learner first explained the algorithm logic in natural language, which led to clear code implementation with detailed comments:

User: 'Yes, queens can be placed in the order of the rows, with one queen per row. At this point, it is necessary to ensure that the queen is not in the same column as any previously placed queens, as well as adhering to the previously mentioned methods of checking that it is not on the left or right diagonal lines'.

ChatGPT: "Thank you very much for your explanation and guidance! I now understand that when placing queens, we can place them row by row, ensuring that the current position is not in the same column as any previously placed queens, and also not on the left or right diagonals. Now I can start thinking about the steps to generate the code ..."

This indicates that the process of instructing the ChatGPT agent helped learners to clarify their thought processes in coding, ultimately contributing to their development of more structured and comprehensible code. Moreover, this learning by teaching approach likely encouraged learners to prioritize the readability and clearness of their code, emphasizing the importance of writing code that is understandable to others (Tan et al., 2024).

However, our findings indicate that students using ChatGPT agents for teaching did not show significant differences in code correctness compared with the CG, with the CG actually achieving slightly higher scores in code correctness. In other words, learners who interacted with ChatGPT agents were likely to produce code with a variety of bugs in the testing phase. This finding prompts us to reconsider the design of ChatGPT's output as a teachable agent. Previous research has shown that allowing learners to discover and correct errors can lead to deeper understanding (Abid et al., 2022; Yang et al., 2024). But in our study, the errors that emerged from ChatGPT's responses were often random and not representative of typical programming misconceptions, which might explain why the EG did not show significant differences in code correctness. Future research should focus on using carefully crafted prompts to make ChatGPT generate code containing specific conceptual misconceptions (Li, 2024). Such intentionally designed errors would be more pedagogically valuable than accidental errors arising from model limitations or hallucinations, as they can be systematically aligned with learning objectives and common student misconceptions in programming education. This approach could ensure that students engage with meaningful and representative programming challenges while maintaining the benefits of the teaching interaction with ChatGPT.

The effect of ChatGPT teachable agent on SRL abilities

For RQ3, this study provides empirical evidence that learning with ChatGPT teachable agent can significantly affect students' SRL in the EG, as measured by the pre-post differences in self-efficacy and cognitive strategies. Overall, we found that students' self-efficacy improved after they teaching ChatGPT teachable agent. As developed by Bandura and Adams (1977), self-efficacy refers to subject-specific confidence in one's ability to succeed in this subject. In our experiment, students' guidance to ChatGPT could mostly successfully generate codes by less attempts to pass the online judgement. In other words, the ChatGPT decreased students' difficulties in programming solution codes when solving the 'N queens puzzle', which improved their confidence in the programming subjects. Moreover, as previous study indicated, the teachable agent allows students to act the role of an expert, boosting their self-esteem and confidence compared with traditional student role (Pareto et al., 2009). Other studies also indicated

that observing the teachable agent's improvement based on their own guidance can act as a form of social modelling (Biswas et al., 2005). Seeing the agent succeed can mirror the learners' own potential for improvement, thereby boosting their belief in their capabilities (Nye & Silverman, 2013). Those factors can be the reasons of students' self-efficacy improvement when they teaching the ChatGPT agent.

We also found that students' cognitive strategies improved more in EG compared with CG. It is worth noting that, unlike previous TAs where students' behaviours and methods of instruction were predefined (eg, Biswas et al., 2005; Matsuda, Keiser, Raizada, et al., 2010), interactions with the ChatGPT agent were different. In the latter case, students needed to plan how to solve tasks themselves. As Zimmerman and Schunk (2011) stated, planning is an essential component of SRL strategies, as it helps students break down a task into manageable parts. The teaching process with the ChatGPT agent could be particularly beneficial as it guides the agent on how to solve a learning task by appropriately breaking it down and formulating a plan. For example, one student broke down the 'N queens' problem by first teaching ChatGPT the fundamental knowledge:

User: "Do you know about backtracking algorithms?"

ChatGPT: "Backtracking algorithm? I've heard of the term, but I'm not entirely clear on what it entails. Could you help me understand this aspect? Specifically, its application in the N-Queens problem. Could you explain the basic idea and steps of the backtracking algorithm, and why it is well-suited for solving the N-Queens problem? Thank you very much!"

User: "The main idea of the backtracking algorithm is to return to the previous step and try a new approach when a certain step has no solution."

ChatGPT: "Oh, I see! The main idea of the backtracking algorithm is that if at any step it's found impossible to proceed, it goes back to the previous step and tries again. I think this approach is quite similar to how humans solve complex problems, and it feels like it could be very useful. Since the backtracking algorithm is suitable for solving the N-Queens problem, what exactly do we need to do next? ..."

User: "Can you first write a rough function based on your understanding?" ...

Previous studies have shown that the ability to divide a learning task into parts and manage these tasks in collaboration with a ChatGPT system is a crucial skill for effective human–AI interaction (Ahmad et al., 2023; Fui-Hoon Nah et al., 2023). Therefore, our findings on the improvement of cognitive strategies suggest that when students teach a teachable ChatGPT agent, they become more engaged and spontaneous in employing detailed SRL strategies. These strategies mirror real-world learning scenarios more closely, where students not only aware their agents' mistakes but also guide them using suitable SRL techniques and adjusting strategies as needed.

Implications

Our study explored the effect of ChatGPT-based TAs on students' learning outcomes, resulting in several implications for educators and researchers: First, our study indicated that the natural language teaching process based on ChatGPT can enhance students' knowledge gains, validating that natural conversation with a teachable agent can be an effective way for students to learn by teaching. Our findings recommend that educators create tasks requiring students to teach or explain concepts to ChatGPT agents using natural language to promote deeper cognitive processes. Second, it is evident that the pedagogical design of using AI as TAs should not focus solely on the accuracy of AI-generated content but also on the learning processes it stimulates. Incorporating scenarios where learners must identify and correct errors, such as purposeful mistakes by the AI, can be beneficial. Our findings suggest that designing future interactions could

focus more on incorporating typical coding errors that students can identify and correct, which may provide opportunities for developing debugging skills and understanding programming concepts. Third, unlike traditional TAs with predefined behaviours, ChatGPT's flexibility allows learners to independently plan and analyse how to solve tasks, akin to real-world scenarios, thus requiring detailed SRL strategies. In other words, the ChatGPT-based teachable agent could be considered one of the methods of SRL intervention that can facilitate students' self-efficacy and more cognitive strategies used. Future research should explore how SRL strategies can be better integrated into the interactions between students and ChatGPT-based TAs. In future research, educators can design tasks where students 'teach' ChatGPT by explaining concepts or solving problems, encouraging them to rephrase and clarify knowledge to foster deeper understanding. ChatGPT can also be programmed to intentionally provide incorrect answers, prompting students to critically evaluate and correct these mistakes, thereby enhancing their critical thinking and problem-solving skills. For example, coding tasks with intentional AI errors can help students practice debugging while solidifying their understanding. Additionally, collaborative learning tasks, where students work in groups to teach and critique ChatGPT, can further promote teamwork and deeper comprehension. Encouraging students to set goals, monitor progress and reflect on their teaching, such as in collaborative programming projects, can make the learning experience more purposeful and effective.

However, it also presents significant challenges that must be addressed for effective and ethical application. One key issue is the inconsistency of ChatGPT's outputs, which can create unpredictable learning experiences and potentially compromise the effectiveness of the LBT approach. Structured prompting frameworks, such as introducing targeted questions or intentional code errors, could standardize interactions and ensure a more controlled and focused learning environment. Additionally, novice learners may struggle to assess and correct AI responses, especially since ChatGPT seldom challenges or contradicts their guidance, potentially reinforcing errors or misconceptions. To mitigate this, scaffolding tools, including rubrics, assessment guidelines and automated feedback systems, should be developed to help students evaluate and improve ChatGPT's responses effectively. Transparency is another critical concern, as ChatGPT's reliance on complex, 'black-box' algorithms can lead to mistrust when system responses deviate from expectations. Enhancing the explainability of agent feedback can help students better understand AI-generated responses, fostering trust and reducing the psychological burden of errors. Moreover, emphasising students' dual roles as both 'teachers' and 'learners' can strengthen engagement and self-efficacy. Finally, there is a risk of misuse, such as repurposing TAs for monitoring or performance assessment, which could diminish the role of teachers and deprive students of essential guidance. Furthermore, there is a concern that interactions with these agents could lead to ethical challenges (Knight et al., 2024), such as over-reliance, potentially hindering the development of essential communication skills that are typically fostered through human interaction (Fan et al., 2025). Teachers must actively oversee interactions, provide guidance and encourage students to value exploration and growth rather than focusing solely on outcomes. By addressing these challenges, TAs like ChatGPT can become powerful tools for enhancing educational experiences.

CONCLUSIONS

Given the growing interest in incorporating generative AI, such as ChatGPT into education to enhance the learning experience, there is still room for investigation into how to empower students to take on a more proactive role in their learning. Learning by teaching is one of an important pedagogical strategies for students to actively focus on proactive

learning, while it faces many limitations like lack of suitable peers. Previous research has shown that using TAs can effectively support this learning process, but these systems often require extensive development and typically lack the ability to facilitate natural language conversation, limiting socialized experiences of LBT. Our study explored the use of ChatGPT as a teachable agent to support students' LBT in programming education. The findings indicate that engaging in natural language conversation with ChatGPT enhances students' knowledge gains and improves their programming skills, particularly in writing readable and logically sound code. However, teaching ChatGPT did not significantly help students develop error-correction skills. Additionally, students' SRL abilities increased, suggesting that the teaching process with ChatGPT provides a better environment for implementing SRL strategies, such as independent planning and problem solving. Overall, the study demonstrates the high potential of ChatGPT as a teachable agent in supporting students' LBT processes, paving the way for future research in this area.

However, our study also meets several limitations. First, our study was conducted with a relatively small sample size of participants, which may bring bias into our results. Future studies should employ larger and more representative sample sizes to enhance the statistical power and generalizability of findings. Second, the study primarily focused on short-term learning outcomes without considering the long-term retention of knowledge and skills. This limitation suggests that further research should examine the enduring impacts of using ChatGPT as a teachable agent over longer periods to determine whether the initial learning gains are sustained and translate into long-term educational benefits. Third, the study did not differentiate between various student demographics, such as age, educational background and gender, which might influence how students interact with and benefit from the ChatGPT-based teachable agent. Future studies should consider these variables to better understand the differential impacts of AI TAs across diverse learner groups. Fourth, we acknowledge that using identical items in pre and post assessments may introduce test-reactivity effects, despite our control measures. Future research might consider developing parallel forms of assessment instruments to address this limitation. Fifth, while our study demonstrated that ChatGPT can exhibit authentic learning characteristics through asking questions as a beginner, we did not systematically investigate students' perceptions of ChatGPT's role as a learner. Future research should explore how students' awareness and understanding of ChatGPT's capabilities might influence their teaching approach and learning outcomes. Finally, while our quantitative results show positive learning outcomes, we lack qualitative data to fully understand how students' perspectives contributed to these outcomes. Future research should incorporate interviews to triangulate the quantitative results with qualitative data.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ETHICS STATEMENT

This study was approved by the university's institutional review board under approval number: IRB00001052-23149.

ORCID

Yang Wei  <https://orcid.org/0000-0002-8187-4011>

REFERENCES

- Abid, A., Yuksekgonul, M., & Zou, J. (2022). Meaningfully debugging model mistakes using conceptual counterfactual explanations. In *International Conference on Machine Learning, PMLR* (pp. 66–88).
- Aflalo, E. (2021). Students generating questions as a way of learning. *Active Learning in Higher Education*, 22(1), 63–75.
- Ahmad, A., Waseem, M., Liang, P., Fahmideh, M., Aktar, M. S., & Mikkonen, T. (2023). Towards human-bot collaborative software architecting with chatgpt. In *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering* (pp. 279–285).
- Andrzejewska, M., Stolińska, A., Błasiak, W., Pęczkowski, P., Rosiek, R., Rożek, B., Sajka, M., & Wcisło, D. (2016). Eye-tracking verification of the strategy used to analyse algorithms expressed in a flowchart and pseudocode. *Interactive Learning Environments*, 24(8), 1981–1995.
- Bandura, A., & Adams, N. E. (1977). Analysis of self-efficacy theory of behavioral change. *Cognitive Therapy and Research*, 1(4), 287–310.
- Baylor, A. L., & Kim, Y. (2005). Simulating instructional roles through pedagogical agents. *International Journal of Artificial Intelligence in Education*, 15(2), 95–115.
- Biswas, G., Jeong, H., Kinnebrew, J. S., Sulcer, B., & Roscoe, R. (2010). Measuring self-regulated learning skills through social interactions in a teachable agent environment. *Research and Practice in Technology Enhanced Learning*, 5(2), 123–152.
- Biswas, G., Katzlberger, T., Bransford, J., & Schwartz, D. (2001). Extending intelligent learning environments with teachable agents to enhance learning. In *Artificial intelligence in education. CiteSeer* (pp. 389–397). IOS Press.
- Biswas, G., Leelawong, K., Belyne, K., Viswanath, K., Schwartz, D., & Davis, J. (2004). Developing learning by teaching environments that support self-regulated learning. In *Intelligent tutoring systems: Proceedings of 7th International Conference, ITS 2004, Maceió, Alagoas, Brazil, August 30–September* (pp. 730–740). Springer.
- Biswas, G., Leelawong, K., Belyne, K., Viswanath, K., Vye, N., Schwartz, D., & Davis, J. (2004). Incorporating self regulated learning techniques into learning by teaching environments. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 26).
- Biswas, G., Leelawong, K., Schwartz, D., Vye, N., & Vanderbilt, T. T. A. G. (2005). Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence*, 19(3–4), 363–392.
- Biswas, G., Roscoe, R., Jeong, H., & Sulcer, B. (2009). Promoting self-regulated learning skills in agent-based learning environments. In *Proceedings of the 17th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education Hong Kong* (pp. 67–74).
- Biswas, G., Segedy, J. R., & Bunchongchit, K. (2016). From design to implementation to practice a learning by teaching system: Betty's brain. *International Journal of Artificial Intelligence in Education*, 26, 350–364.
- Blair, K., Schwartz, D. L., Biswas, G., & Leelawong, K. (2007). Pedagogical agents for learning by teaching: Teachable agents. *Educational Technology*, 47(1), 56–61.
- Boekaerts, M., Otten, R., & Voeten, R. (2003). Examination performance: Are student's causal attributions school-subject specific? *Anxiety, Stress, and Coping*, 16(3), 331–342.
- Borm, G. F., Fransen, J., & Lemmens, W. A. (2007). A simple sample size formula for analysis of covariance in randomized clinical trials. *Journal of Clinical Epidemiology*, 60(12), 1234–1238.
- Börstler, J., Bennin, K. E., Hooshangi, S., Jeuring, J., Keuning, H., Kleiner, C., MacKellar, B., Duran, R., Störrle, H., Toll, D., & van Assema, J. (2023). Developers talking about code quality. *Empirical Software Engineering*, 28(6), 128.
- Börstler, J., Bin Ali, N., Svensson, M., & Petersen, K. (2023). Investigating acceptance behavior in software engineering—Theoretical perspectives. *Journal of Systems and Software*, 198, 111592.
- Bredeweg, B., Gómez-Pérez, A., André, E., & Salles, P. (2009). Dynalearnengaging and informed tools for learning conceptual system knowledge. In *2009 AAAI Fall Symposium Series*.
- Brown, T. B. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Chase, C. C., Chin, D. B., Oppezzo, M. A., & Schwartz, D. L. (2009). Teachable agents and the protégé effect: Increasing the effort towards learning. *Journal of Science Education and Technology*, 18, 334–352.
- Chen, Z., Zhou, K., Zhang, B., Gong, Z., Zhao, W. X., & Wen, J. R. (2023). Chatcot: Tool-augmented chain-of-thought reasoning on chat-based large language models. *arXiv preprint arXiv:2305.14323*.
- Chi, M. T., & Wylie, R. (2014). The icap framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist*, 49(4), 219–243.
- Chin, C., & Brown, D. E. (2002). Student-generated questions: A meaningful aspect of learning in science. *International Journal of Science Education*, 24(5), 521–549.
- Chin, D. B., Dohmen, I. M., Cheng, B. H., Oppezzo, M. A., Chase, C. C., & Schwartz, D. L. (2010). Preparing students for future learning with teachable agents. *Educational Technology Research and Development*, 58, 649–669.

- Debbané, A., Lee, K. J., Tse, J., & Law, E. (2023). Learning by teaching: Key challenges and design implications. *Proceedings of the ACM on Human-Computer Interaction*, 7, 1–34.
- Duran, D. (2017). Learning-by-teaching. Evidence and implications as a pedagogical mechanism. *Innovations in Education and Teaching International*, 54(5), 476–484.
- Fan, Y., Tang, L., Le, H., Shen, K., Tan, S., Zhao, Y., Shen, Y., Li, X., & Gašević, D. (2025). Beware of metacognitive laziness: Effects of generative artificial intelligence on learning motivation, processes, and performance. *British Journal of Educational Technology*, 56(2), 489–530.
- Fui-Hoon Nah, F., Zheng, R., Cai, J., Siau, K., & Chen, L. (2023). Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration. *Journal of Information Technology Case and Application Research*, 25(3), 277–304.
- Han, J. H., Shubeck, K., Shi, G. H., Hu, X. E., Yang, L., Wang, L. J., Zhao, W., Jiang, Q., & Biswas, G. (2021). Teachable agent improves affect regulation. *Educational Technology & Society*, 24(3), 194–209.
- Jun, Y. (2003). Facilitating mathematical learning with a peer tutoring system: Lessons learned. *Journal of Computers in Mathematics and Science Teaching*, 22(1), 75–92.
- Junprung, E. (2023). Exploring the intersection of large language models and agent-based modeling via prompt engineering. *arXiv preprint arXiv:2308.07411*.
- Kazemitabaar, M., Hou, X., Henley, A., Ericson, B. J., Weintrop, D., & Grossman, T. (2023). How novices use LLM-based code generators to solve CS1 coding tasks in a self-paced learning environment. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research* (pp. 1–12).
- Kazemitabaar, M., Ye, R., Wang, X., Henley, A. Z., Denny, P., Craig, M., & Grossman, T. (2024). Codeaid: Evaluating a classroom deployment of an LLM-based programming assistant that balances student and educator needs. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (pp. 1–20).
- Kim, G., Baldi, P., & McAleer, S. (2024). Language models can solve computer tasks. *Advances in Neural Information Processing Systems*, 36, 39648–39677.
- Kim, Y., & Baylor, A. L. (2016). Research-based design of pedagogical agent roles: A review, progress, and recommendations. *International Journal of Artificial Intelligence in Education*, 26, 160–169.
- Kim, Y., & Lim, J. H. (2013). Gendered socialization with an embodied agent: Creating a social and affable mathematics learning environment for middle-grade females. *Journal of Educational Psychology*, 105(4), 1164–1174.
- King, A., Staffieri, A., & Adelgais, A. (1998). Mutual peer tutoring: Effects of structuring tutorial interaction to scaffold peer learning. *Journal of Educational Psychology*, 90(1), 134–152.
- Kinnebrew, J. S., Biswas, G., Sulcer, B., & Taylor, R. S. (2013). Investigating self-regulated learning in teachable agent environments. In *International handbook of metacognition and learning technologies* (pp. 451–470). Springer.
- Knight, S., Viberg, O., Mavrikis, M., Kovanović, V., Khosravi, H., Ferguson, R., Corrin, L., Thompson, K., Major, L., Lodge, J., Hennessy, S., & Cukurova, M. (2024). Emerging technologies and research ethics: Developing editorial policy using a scoping review and reference panel. *PLoS One*, 19(10), e0309715.
- Kobayashi, K. (2019). Learning by preparing-to-teach and teaching: A meta-analysis. *Japanese Psychological Research*, 61(3), 192–203.
- Kong, A., Zhao, S., Chen, H., Li, Q., Qin, Y., Sun, R., Zhou, X., Wang, E., & Dong, X. (2023). Better zero-shot reasoning with role-play prompting. *arXiv preprint arXiv:2308.07702*.
- Lachner, A., Backfisch, I., Hoogerheide, V., Gog, T., & Renkl, A. (2019). Timing matters! Explaining between study phases enhances students' learning. *Journal of Educational Psychology*, 112. <https://doi.org/10.1037/edu0000396>
- Lachner, A., Hoogerheide, V., van Gog, T., & Renkl, A. (2022). Learning-by-teaching without audience presence or interaction: When and why does it work? *Educational Psychology Review*, 34(2), 575–607.
- Leelawong, K., & Biswas, G. (2008). Designing learning by teaching agents: The betty's brain system. *International Journal of Artificial Intelligence in Education*, 18(3), 181–208.
- Li, M. Z. (2024). Using prompt engineering to enhance stem education. In *2024 IEEE Integrated STEM Education Conference (ISEC)* (pp. 1–2). IEEE.
- Liao, X., Zhang, X., Wang, Z., & Luo, H. (2024). Design and implementation of an ai-enabled visual report tool as formative assessment to promote learning achievement and self-regulated learning: An experimental study. *British Journal of Educational Technology*, 55(3), 1253–1276.
- Lin, B. Y., Fu, Y., Yang, K., Brahman, F., Huang, S., Bhagavatula, C., Ammanabrolu, P., Choi, Y., & Ren, X. (2024). Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. *Advances in Neural Information Processing Systems*, 36, 23813–23825.
- Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., Wu, Z., Zhao, L., Zhu, D., Li, X., Qiang, N., Shen, D., Liu, T., & Ge, B. (2023). Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, 1, 100017.
- Lo, C. K. (2023). What is the impact of chatgpt on education? A rapid review of the literature. *Education Sciences*, 13(4), 410.

- Luo, J. J. (2024). How does GenAI affect trust in teacher-student relationships? Insights from students' assessment experiences. *Teaching in Higher Education*, 1–16. <https://doi.org/10.1080/13562517.2024.2341005>
- Markel, J. M., Opferman, S. G., Landay, J. A., & Piech, C. (2023). Gpteach: Interactive ta training with gpt-based students. In *Proceedings of the Tenth ACM Conference on Learning@ Scale* (pp. 226–236).
- Matsuda, N. (2022). Teachable agent as an interactive tool for cognitive task analysis: A case study for authoring an expert model. *International Journal of Artificial Intelligence in Education*, 32(1), 48–75.
- Matsuda, N., Cohen, W. W., Koedinger, K. R., Keiser, V., Raizada, R., Yarzebinski, E., Watson, S. P., & Stylianides, G. (2012). Studying the effect of tutor learning using a teachable agent that asks the student tutor for explanations. In *2012 IEEE Fourth International Conference on Digital Game and Intelligent Toy Enhanced Learning* (pp. 25–32). IEEE.
- Matsuda, N., Cohen, W. W., Koedinger, K. R., Stylianides, G., Keiser, V., & Raizada, R. (2010). Tuning cognitive tutors into a platform for learning-by-teaching with SimStudent technology. In *1st APLEC Workshop Proceedings*.
- Matsuda, N., Keiser, V., Raizada, R., Tu, A., Stylianides, G., Cohen, W. W., & Koedinger, K. R. (2010). Learning by teaching SimStudent: Technical accomplishments and an initial use with students. In *Intelligent tutoring systems: 10th International Conference, ITS 2010, Pittsburgh, PA, June 14–18, Proceedings, Part I 10* (pp. 317–326). Springer.
- Matsuda, N., Sekar, V. P. C., & Wall, N. (2018). Metacognitive scaffolding amplifies the effect of learning by teaching a teachable agent. In *Artificial intelligence in education: 19th International Conference, AIED 2018, London, UK, June 27–30, Proceedings, part I* (Vol. 19, pp. 311–323). Springer.
- Matsuda, N., Yarzebinski, E., Keiser, V., Raizada, R., Cohen, W. W., Stylianides, G. J., & Koedinger, K. R. (2013). Cognitive anatomy of tutor learning: Lessons learned with SimStudent. *Journal of Educational Psychology*, 105(4), 1152–1163.
- Nasir, R., & Lee, W. (2014). Knowledge building and knowledge creation: Theory, pedagogy, and technology. In *The Cambridge handbook of the learning sciences* (pp. 687–706). Cambridge University Press.
- Nelson-Le Gall, S. (1981). Help-seeking: An understudied problem-solving skill in children. *Developmental Review*, 1(3), 224–246.
- Nye, B. D., & Silverman, B. G. (2013). Social learning and adoption of new behavior in a virtual agent society. *Presence*, 22(2), 110–140.
- Ogan, A., Finkelstein, S., Mayfield, E., D'Adamo, C., Matsuda, N., & Cassell, J. (2012). “Oh dear stacy!”: Social interaction, elaboration, and learning with teachable agents. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'12* (pp. 39–48). Association for Computing Machinery.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., & Schulman, J. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 27730–27744.
- Packer, C., Fang, V., Patil, S. G., Lin, K., Wooders, S., & Gonzalez, J. E. (2023). MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*.
- Panadero, E. (2017). A review of self-regulated learning: Six models and four directions for research. *Frontiers in Psychology*, 8, 422.
- Pareto, L. (2014). A teachable agent game engaging primary school children to learn arithmetic concepts and reasoning. *International Journal of Artificial Intelligence in Education*, 24, 251–283.
- Pareto, L., Arvemo, T., Dahl, Y., Haake, M., & Gulz, A. (2011). A teachable agent arithmetic game's effects on mathematics understanding, attitude and self-efficacy. In *Artificial intelligence in education: 15th International Conference, AIED 2011, Auckland, New Zealand, June 28–July 15, 2011* (pp. 247–255). Springer.
- Pareto, L., Schwartz, D. L., & Svensson, L. (2009). Learning by guiding a teachable agent to play an educational game. In *Artificial intelligence in education* (pp. 662–664). IOS Press.
- Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (pp. 1–22).
- Pintrich, P. (2000). The role of goal orientation in self-regulated learning. In *Handbook of self-regulation* (pp. 451–502). Academic Press.
- Pintrich, P. R., & De Groot, E. V. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psychology*, 82(1), 33–40.
- Pintrich, P. R., Smith, D. A., Garcia, T., & McKeachie, W. J. (1993). Reliability and predictive validity of the motivated strategies for learning questionnaire (MSLQ). *Educational and Psychological Measurement*, 53(3), 801–813.
- Puustinen, M., & Pulkkinen, L. (2001). Models of self-regulated learning: A review. *Scandinavian Journal of Educational Research*, 45(3), 269–286.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 9.
- Roscoe, R. D., & Chi, M. T. (2007). Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions. *Review of Educational Research*, 77(4), 534–574.

- Roscoe, R. D., Segedy, J. R., Sulcer, B., Jeong, H., & Biswas, G. (2013). Shallow strategy development in a teachable agent environment designed to support self-regulated learning. *Computers & Education*, 62, 286–297.
- Ross, S. I., Martinez, F., Houde, S., Muller, M., & Weisz, J. D. (2023). The programmer's assistant: Conversational interaction with a large language model for software development. In *Proceedings of the 28th International Conference on Intelligent User Interfaces* (pp. 491–514).
- Schwartz, D. L., Chase, C., Chin, D. B., Oppezzo, M., Kwong, H., Okita, S., Biswas, G., Roscoe, R., Jeong, H., & Wagster, J. (2009). Interactive metacognition: Monitoring and regulating a teachable agent. In *Handbook of metacognition in education* (pp. 340–358). Routledge.
- Silvervarg, A., Wolf, R., Blair, K. P., Haake, M., & Gulz, A. (2021). How teachable agents influence students' responses to critical constructive feedback. *Journal of Research on Technology in Education*, 53(1), 67–88.
- Song, M. K., & Ward, S. E. (2015). Assessment effects in educational and psychosocial intervention trials: An important but often-overlooked problem. *Research in Nursing & Health*, 38(3), 241–247.
- Spruce, R., & Bol, L. (2015). Teacher beliefs, knowledge, and practice of self-regulated learning. *Metacognition and Learning*, 10, 245–277.
- Tan, J., Wu, L., & Ma, S. (2024). Collaborative dialogue patterns of pair programming and their impact on programming self-efficacy and coding performance. *British Journal of Educational Technology*, 55(3), 1060–1081.
- Terrace, H. S. (2005). Metacognition and the evolution of language. In *The missing link in cognition: Origins of self-reflective consciousness* (pp. 84–115). Oxford University Press.
- Walker, E., Rummel, N., & Koedinger, K. R. (2014). Adaptive intelligent support to improve peer tutoring in algebra. *International Journal of Artificial Intelligence in Education*, 24, 33–61.
- Yang, S., Baird, M., O'Rourke, E., Brennan, K., & Schneider, B. (2024). Decoding debugging instruction: A systematic literature review of debugging interventions. *ACM Transactions on Computing Education*, 24(4), 1–44.
- Zhao, G., Ailiya, & Shen, Z. (2012). Learning-by-teaching: Designing teachable agents with intrinsic motivation. *Journal of Educational Technology & Society*, 15(4), 62–74.
- Zhou, W., Jiang, Y. E., Cui, P., Wang, T., Xiao, Z., & Hou, Y. (2023). Recurrentgpt: Interactive generation of (arbitrarily) long text. *arXiv preprint arXiv:2305.13304*.
- Zimmerman, B. (2000). Attaining self-regulation: A social cognitive perspective. In *Self-regulation: Theory, research, and applications*. Academic Press.
- Zimmerman, B. J. (2002). Becoming a self-regulated learner: An overview. *Theory Into Practice*, 41(2), 64–70.
- Zimmerman, B. J., & Schunk, D. H. (2011). Self-regulated learning and performance: An introduction and an overview. In *Handbook of selfregulation of learning and performance* (pp. 15–26). Routledge. <https://doi.org/10.4324/9780203839010>.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Chen, A., Wei, Y., Le, H., & Zhang, Y. (2025). Learning by teaching with ChatGPT: The effect of teachable ChatGPT agent on programming education. *British Journal of Educational Technology*, 00, 1–22. <https://doi.org/10.1111/bjet.70001>