Humanities & Social Sciences Communications

ARTICLE

Check for updates

https://doi.org/10.1057/s41599-024-03991-6 OPEN

Investigating students' programming behaviors, interaction qualities and perceptions through prompt-based learning in ChatGPT

Dan Sun[1], Azzeddine Boudouaia[2]✉, Junfeng Yang[3] & Jie Xu[4]

ChatGPT has proven to facilitate computer programming tasks through the strategic use of prompts, which effectively steer the interaction with the language model towards eliciting relevant information. However, the impact of specifically designed prompts on programming learning outcomes has not been rigorously examined through empirical research. This study adopted a quasi-experimental framework to investigate the differential effects of prompt-based learning (PbL) versus unprompted learning (UL) conditions on the programming behaviors, interaction qualities, and perceptions of college students. The study sample consisted of 30 college students who were randomly assigned to two groups. A mixed-methods approach was employed to gather multi-faceted data. Results revealed notable distinctions between the two learning conditions. First, the PbL group students frequently engaged in coding with Python and employed debugging strategies to verify their work, whereas their UL counterparts typically transferred Python code from PyCharm into ChatGPT and posed new questions within ChatGPT. Second, PbL participants were inclined to formulate more complex queries independently, prompted by the guiding questions, and consequently received more precise feedback from ChatGPT compared to the UL group. UL students tended to participate in more superficial-level interactions with ChatGPT, yet they also obtained accurate feedback. Third, there were noticeable differences in perception observed before and after the ChatGPT implementation, UL group reported a more favorable perception in the perceived ease of use in the pre-test, while the PbL group experienced an improvement in their mean scores for perceived usefulness, ease of use, behavioral intention to utilize, and a significant difference regarding the attitude towards utilizing ChatGPT. Specifically, the use of structured output and delimiters enhanced learners' understanding of problem-solving steps and made learning more efficient with ChatGPT. Drawing on these outcomes, the study offers recommendations for the incorporation of ChatGPT into future instructional designs, highlighting the structured prompting benefits in enhancing programming learning experience.

[1] Chinese Education Modernization Research Institute of Hangzhou Normal University (Zhejiang Provincial Key Think Tank), Hangzhou, China. [2] School of Foreign Languages, Southwest Jiaotong University, Chengdu, China. [3] Jinghengyi School of Education, Hangzhou Normal University, Hangzhou, China. [4] College of Education, Zhejiang University, Hangzhou, China. ✉email: azzeddine-link@hotmail.com

Introduction

In today's tech-driven culture, programming education is becoming increasingly vital (Liu et al. 2023; Sun et al. 2022). The reason behind this is its widely acknowledged importance in enabling the creation of scripts and programs capable of efficiently handling repetitive or time-consuming tasks (Ou et al. 2023). Because of this, students can focus more intently on the challenging and innovative aspects of their work, which increases productivity. Students who take programming classes will be more prepared to succeed in today's data-driven and technologically advanced workplace (Dengler and Matthes 2018). It gives students the tools they need to be adaptable, competent in using computers, and problem solvers, which makes them more marketable in the job market (Nouri et al. 2020). The pursuit of programming education's benefits, however, is not without its challenges. Several obstacles may arise for students who are actively participating in programming education.

Recent studies have shown that programming is ideally aligned with the fast rise of artificial intelligence (Yilmaz and Karaoglan Yilmaz 2023a). The alignment may be attributed to the many benefits that students get from artificial intelligence resources (Andersen et al. 2022). Computer programming abilities, such as coding ability and task understanding, may be significantly improved with the aid of artificial intelligence. Recently, students have begun using ChatGPT in their programming learning (Surameery and Shakor 2023). ChatGPT has had a substantial influence on the learning process as a result of its capacity to assist learners by giving explanations, responding to queries, and offering help on a broad variety of topics (Gayed et al. 2022; Jeon and Lee 2023). Even for those who are experiencing difficulties in their educational journeys, ChatGPT has the potential to improve accessibility because of its capacity to provide information and help to users who come from a variety of learning backgrounds (Kohnke et al. 2023; Tam 2023). When it comes to the more tedious aspects of learning, ChatGPT is a great value tool that helps and enhances learners' efficiency (Ahmed 2023). ChatGPT's extensive knowledge base and linguistic fluency allow it to aid users in discovering information or having amusing discussions (Kohnke et al. 2023; Tam 2023).

Because of its remarkable capacity to comprehend and generate natural language that is indistinguishable from that of a genuine human speaker, ChatGPT stands out as a potent automated language processing tool (Dwivedi et al. 2023; Kohnke et al. 2023). ChatGPT, when used in conjunction with scaffolding and activity creation, might greatly enhance learning (Gayed et al. 2022; Jeon and Lee, 2023). To generate responses that resemble human experts while answering questions and addressing different topics, ChatGPT combines unsupervised pre-training with supervised fine-tuning (Dwivedi et al. 2023). During a discussion, it could answer many questions. The model gets data during the session and processes it to provide replies (Gayed et al. 2022). Not only does the system not save or retrieve information from previous sessions or discussions, but each interaction stands on its own and is independent of any previous state. This system cannot learn from its experiences and retain information about previous encounters, in contrast to those that have real in-context learning. To make it look livelier and more participatory, the conversational agent could offer questions to steer the conversation. However, there is a lack of supporting evidence for its results. The current state of affairs at ChatGPT puts it at risk of being seen as the last word in knowledge, as it states a single truth without providing enough evidence or suitable qualifiers (Cooper, 2023; Kohnke et al. 2023; Rawas, 2023). In this instance, students had a difficulty in effectively using ChatGPT to get essential feedback for their programming education. Several students use ChatGPT haphazardly and choose its responses without concern for proper sequencing, on the assumption that ChatGPT would provide accurate output regardless of input quality. As a result, students have limitations while using ChatGPT. Problems that students face when learning to code in college could have a negative impact on their whole academic career. Sun et al. (2021a) and Lu et al. (2017) both emphasize the need to provide college students with significant support throughout their programming education.

In this respect, it is necessary to construct scaffolding by integrating prompts to adequately aid ChatGPT in performing productive or well-established answers. Brown et al. (2020) found that prompt-based learning, a less technical no-code method to "instruct" big language models on a specific task, has recently shown encouraging outcomes. In natural language, prompts may be either brief instructions with examples (Yin et al. 2019) or more detailed descriptions of the job at hand. Regarding this matter, the usefulness of ChatGPT for programming education may be called into question if its usage differs when a prompt is included compared to when it is not. The most effective approach for using ChatGPT to enhance students' learning of programming is still uncertain (Sun et al. 2024a). In this respect, it seems worthwhile to investigate the effectiveness of using prompts while using ChatGPT in programming learning.

Previous research had demonstrated that acceptance has a substantial impact on learning performances (Liu and Ma 2023; Ofosu-Ampong et al. 2023). In order to gain a more comprehensive understanding of how individuals adopt and employ new technologies, such as ChatGPT, researchers have employed the Technology Acceptance Model (TAM), which was developed by Davis (1989). Sánchez and Hueros (2010) have proven the reliability of the TAM as a predictor of consumer adoption of new technologies. The TAM posits that users' assessments of a technology's usefulness and ease of use are critical determinants of their willingness to adopt and implement it. Furthermore, previous research and TAM have demonstrated that users' attitudes and intentions also significantly influence their willingness to utilize technology for educational purposes (Yi et al. 2016; Sun et al. 2024a). The acceptance of prompts in ChatGPT is a novel area of investigation that has not yet garnered researchers' attention. It is an essential determinant of ChatGPT's efficacy in facilitating student programming learning. Researchers have used Davis' (1989) technology acceptance model to look at how students interact with AI-based learning technologies (Xie et al. 2023) in the area of programming learning that works with prompts (Sun et al. 2024a). This implies that TAM can be used to study how students accept PBL in ChatGPT. Thongkoo et al. (2020) show that the TAM is a good way to predict how people will use technology to learn programming. However, there is not much research on how students feel about using prompts when learning programming with ChatGPT, even though Ofosu-Ampong et al. (2023) and Strzelecki (2023) say that some studies have looked at how likely students are to use ChatGPT in the classroom. As a result, it is imperative to investigate whether the use of prompts affects students' perceptions of ChatGPT in programming education, using TAM as a theoretical foundation.

To address the above-mentioned gaps, this study aimed to explore the impact of prompts on college students' programming behaviors, interaction qualities, and perceptions while using ChatGPT in programming learning. This was achieved by comparing two kinds of classes: prompt-based learning (PbL) and unprompted learning (UL). As a result, this study attempted to answer the following research questions:

RQ 1. How did the impact of PbL while using ChatGPT on learners' programming behaviors differ from the impact of UL?

RQ 2. How did the impact of PbL on learners' interaction quality with ChatGPT differ from the impact of UL?

RQ 3. How did the impact of PbL on learners' perceptions toward ChatGPT differ from the impact of UL?

Insights gained from this study may help policymakers improve and expand access to ChatGPT for programming courses at universities. This study may be useful for curriculum developers and teachers in China as they work to improve programming education. Policymakers, curriculum developers, and teachers from many countries may gain valuable insight into their own programming education system's strengths and weaknesses by observing the results of this study. The inclusion of prompts in ChatGPT for programming instruction is an important issue, and this study has the potential to provide new insights into this pressing subject. The findings of this study might contribute to the continuing discussion regarding programming education throughout the world and provide useful information for the developers of ChatGPT. This study has shed light on prospective students' perceptions regarding the use of ChatGPT prompts for programming studies, and it may substantially improve education stakeholders' knowledge.

Literature review

Advancing programming education with ChatBots. To become experts in digital technology and innovators, students need a solid foundation in computer programming (Balanskat and Engelhardt, 2015). Research and development in AI pave the way for smarter technology, such as chatbots, to improve programming education and speed up improvements to talent training methods. Besides, programming and educational breakthroughs have had an impact on AI. In this regard, the next generation must acquire coding skills and digital competencies in light of the pervasiveness of AI (Sun et al. 2021b). The area of artificial intelligence is increasingly recognizing programming education as a very practical kind of learning (Liu et al. 2021). In this vein, coding skills hold significant value in this field (Sun et al. 2021b).

Yilmaz and Karaoglan Yilmaz (2023a) have designed an assortment of artificial intelligence tools, including chatbots, interactive coding platforms, and virtual instructors, to augment students' engagement and accessibility in programming education. Students who are in the process of learning programming will discover these resources beneficial as they endeavor to enhance their understanding of the foundational principles of programming (Tian et al. 2023). Utilizing artificial intelligence tools, students can generate code that is more easily understood, this feature enables individuals to promptly recognize and rectify errors, thereby expediting their educational progression (Yilmaz and Karaoglan Yilmaz 2023b). Providing students with the tools and resources necessary to investigate programming concepts autonomously is an additional aim of these resources, which is to encourage students to pursue their studies independently. This approach fosters self-directed learning and permits every student to acquire coding skills in a distinct manner and at an individualized pace (Eason et al. 2023; Yilmaz and Karaoglan Yilmaz 2023a).

In recent years, there has been a significant increase in the number of students studying programming who have embraced ChatGPT as a valuable tool. This AI model has unique qualities that set it apart from other programming tools. Engaging in conversations with students using natural language allows for personalized explanations, as emphasized in recent studies (Dwivedi et al. 2023; Gayed et al. 2022; Jeon and Lee 2023). Phung et al. (2023) highlighted the progression from ChatGPT to GPT-4, demonstrating its remarkable performance in a range of programming tasks. Based on the findings, it is evident that GPT has demonstrated a remarkable ability to perform at the same level as human instructors across various programming educational tasks. These tasks include program repair, hint generation, grading feedback, peer programming, task development, and contextualized explanation.

ChatGPT's adaptability is clear as it enables individuals to pursue self-directed learning, surpassing the boundaries of traditional educational settings. Examining past discussions or topics can prove to be a valuable asset for self-directed learners seeking to improve their understanding (Gayed et al. 2022). ChatGPT's interface carefully caters to the needs of students with varying levels of programming knowledge, ensuring accessibility and inclusivity for all. Emphasizing usability enhances the overall educational experience (Tian et al. 2023). Even beginners in coding can find value in ChatGPT's intuitive interface and useful support in tackling programming obstacles (Surameery and Shakor 2023). The research conducted by Kashefi and Mukerji (2023) clearly demonstrates ChatGPT's proficiency in numerical problem-solving. This study explores the effectiveness of ChatGPT in addressing numerical algorithms, demonstrating its capacity to generate, debug, enhance, finish, and modify code in different programming languages for a range of numerical problems. In their study, Ouh et al. (2023) offer valuable insights into the effectiveness of ChatGPT in facilitating coding solutions for Java programming tasks within the context of an undergraduate course. Nevertheless, the study highlights the difficulties that ChatGPT encounters when dealing with intricate coding tasks that demand thorough explanations or the integration of class files, resulting in less precise results. The results indicate that ChatGPT has demonstrated remarkable aptitude for generating Java programming solutions that are remarkably lucid and structured. Moreover, the model possesses the capability to generate alternative solutions that exhibit remarkable memory efficiency. Within the wider scope of programming education, these studies emphasize that ChatGPT provides more than just problem-solving capabilities. This resource is extremely valuable for students as they navigate through programming challenges and explore various problem-solving approaches. Understanding the constraints of ChatGPT can assist educators in creating coding assignments that minimize the potential for cheating while still maintaining their effectiveness as assessment tools.

In addition, the literature emphasizes the significant impact of ChatGPT in making complicated topics more accessible by incorporating real-life examples or similar situations (Kohnke et al. 2023; Sun et al. 2024a; Tam 2023). In their study, Avila-Chauvet et al. (2023) showed how ChatGPT can be used for programming assistance. They showcase the real-world implementation of building an online behavioral task using JavaScript, HTML, and CSS. While the AI cannot fully replace programmers, empirical evidence suggests that OpenAI ChatGPT can significantly reduce programming time and offer a wider range of programming options than programmers. In a recent study, Biswas (2022) delves into the potential for enhancing comprehension by improving ChatGPT's computer programming skills. The model's exceptional ability to perform various programming tasks while significantly reducing programming time and providing a wider range of programming options is emphasized. The results showed that ChatGPT can execute a wide range of programming-related tasks and has been trained on a diverse set of texts. Completing and correcting code, predicting and suggesting code snippets, resolving automated syntactic errors, providing recommendations for code optimization and refactoring, generating documents and chatbots, responding to technical inquiries, and generating missing code are among these responsibilities. Besides, ChatGPT proved capable of aiding users in comprehending intricate concepts and technologies, locating pertinent resources, and diagnosing and resolving technical issues through the provision of explanations, examples, and direction. Students who require assistance with sophisticated code segments or grammatical errors may find ChatGPT to be incredibly useful. This resource offers extensive advice on correcting errors and optimizing code (Vukojičić and Krstić, 2023). ChatGPT engages students in thought-provoking dialogues and delivers succinct elucidations on specific topics, thereby captivating them. ChatGPT is an excellent tool for learners, offering clear explanations and helpful guidance to greatly improve their grasp of programming fundamentals.

Prompt utilization in ChatBots study. An often-observed method of manipulating the large language model's (LLM) output in chatbots is through the use of explicit queries or instructions (McTear et al. 2023). Providing a prompt is one method to direct the conversation and elicit relevant information from a large language model such as chatbot (White et al. 2023). Recent developments in LLM systems have demonstrated exceptional capabilities for zero-shot learning (Wei et al. 2021). These approaches, also known as prompt-based techniques, have demonstrated efficacy in various domains and tasks (Pryzant et al. 2023; Xu et al. 2022), all without the need for additional training or refinement. This substantially increases the potential for the development of educational chatbots. Researchers have utilized the Reinforcement Learning from Human Feedback (RLHF) framework to enhance the security and dependability of chatbots, including ChatGPT (Ouyang et al. 2022). Implementing a prompt-based approach enables the development of chatbots capable of serving diverse instructional objectives. Through authentic and interactive engagement with Chatbots, students possess the ability to express their viewpoints and direct the discourse in a way that piques their curiosity (Babe et al. 2023; McTear et al. 2023). Requesting information regarding a specific topic may prompt chatbot to provide details in response to an inquiry like "Tell me more about [TOPIC]." As a result, the chatbot provides an extensive array of information and diverse viewpoints pertaining to the topic that has been selected. Incorporating an inquiry into chatbot, such as "I am encountering some difficulties with [SPECIFIC ISSUE]" to ensure that it is capable of providing technical assistance. Individuals will experience enhanced clarity in articulating their distinct concerns. It appears that this inquiry would prompt the chatbot to consider potential solutions to the issue that has been delineated.

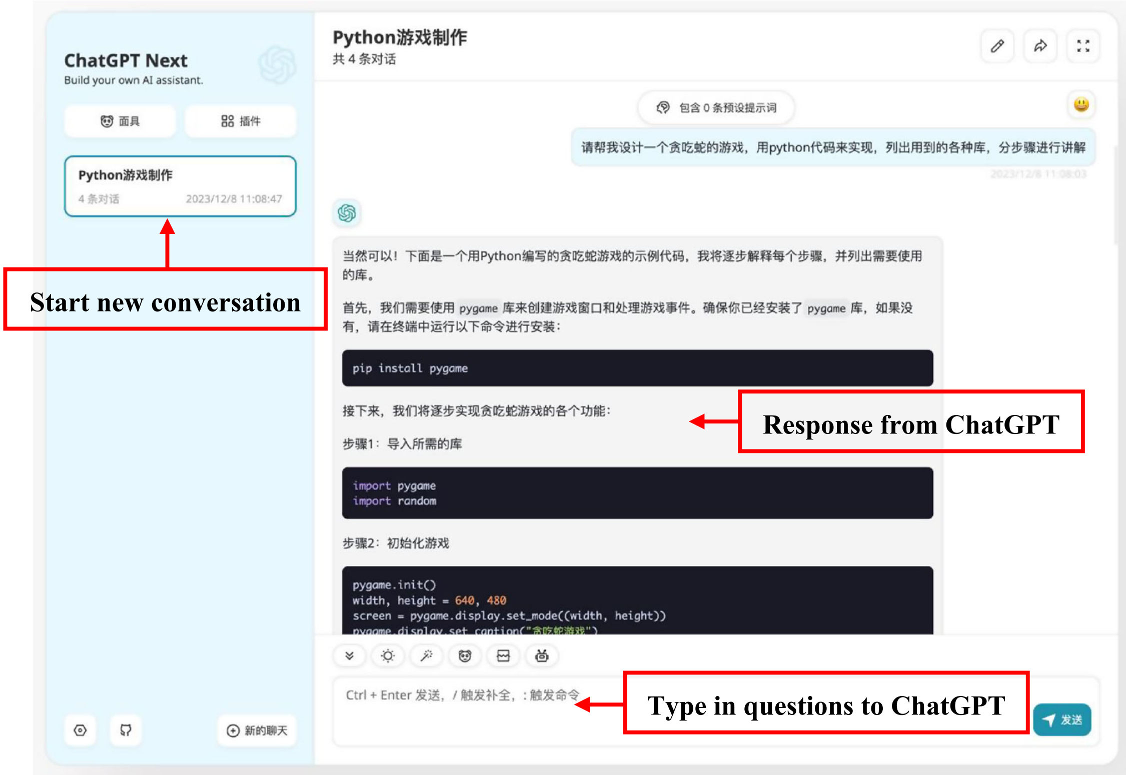
Utilizing extremely large pre-trained language models can be an efficient method in the computer science field, as demonstrated by prompt-based learning (Liu et al. 2021). Constructing a prompt template that describes the downstream task is essential when using a pre-trained language model for prediction tasks. Next, the original input is inserted into the input slot in the prompt template (Schick and Schutze 2021). In this study, we refer to the method of learners using prompt templates to communicate with LLM as prompt-based learning, aligning with the computer science rationale of prompt-based learning approach in programming education. Prompts can enhance chatbot performance in a learning context by allowing students to express their opinions and questions, thereby guiding the dialogue towards their interests. Utilizing prompts is a practical method for directing conversations with chatbots and ensuring that they provide accurate and useful responses (Ekin 2023; McTear et al. 2023). Numerous benefits accompany the use of chatbot prompts. Prompts serve primarily to direct and furnish the chatbot with relevant information. In order to direct the discourse appropriately and obtain valuable responses, it is advisable for users to furnish precise instructions or pose comprehensive inquiries (Mollick and Mollick 2023). Users may also obtain targeted and precise information through the use of prompts (Denny et al. 2023).

In their study, White et al. (2023) introduced a framework that facilitates the creation of problem-domain-relevant prompts. The purpose of this framework is to optimize the performance of LLM responses through the provision of an assortment of inquiries that have been empirically validated. Through the integration and utilization of patterns in the construction of prompts, the framework endeavors to enhance the academic achievements of students who employ LLMs. An area that is becoming increasingly popular is the study of the questions that students ask themselves when employing LLMs. In the course of their investigation, Babe et al. (2023) assembled a reliable dataset consisting of 1,749 queries covering a wide range of 48 distinct topics. Students who have recently been instructed in Python present a diverse array of inquiries in the dataset. The inquiries encompass a wide range of Python programming topics and offer valuable insights into the sorts of inquiries that students frequently pose. All individuals have access to this dataset for the purposes of LLM benchmarking and tool development. In their study, Denny et al. (2023) proposed "Prompt Problem" as a novel educational concept to teach students how to construct prompt for LLMs in an efficient manner. Furthermore, they developed an instrument called Promptly facilitates the assessment of code generated by prompts. The researchers examined the use of Promptly in a first-year Python programming course through a field investigation involving 54 participants. Students highly favored Promptly in the first-year Python programming course as it fostered the development of computational thinking skills and instructed them in novel programming principles.

Moreover, Fulford and Andrew Ng developed the course "ChatGPT Prompt Engineering for Developers" in partnership with OpenAI. The main goal is to help developers optimize the utilization of LLMs. This course integrates the latest research findings on effective techniques for utilizing prompts in the current LLM models. ChatGPT Prompt Engineering for Developers is a great resource for individuals who are just starting out in the field. A basic understanding of Python is enough. On the other hand, it is suitable for experienced machine learning engineers who are interested in delving into the cutting-edge field of prompt engineering and utilizing LLMs. They can gain the necessary skills to effectively utilize a powerful LLM for quickly developing innovative and reliable applications. With the help of the OpenAI API, students can easily create features that can generate innovative ideas and add value in ways that were once costly, highly specialized, or simply out of reach. Isa Fulford from OpenAI and Andrew Ng from Deep Learning teach this course. AI can provide insights into how LLMs work, provide tips for effective prompt engineering, and showcase the use of LLM APIs in various applications. These applications include summarization (such as condensing user reviews), inference (such as sentiment analysis and topic extraction), text transformation (such as translation and spelling/grammar correction), and expansion (such as automated email composition). In addition, students can gain an understanding of two key concepts for creating effective prompts, as well as the systematic approach to designing prompts of high quality. In addition, students can develop their skills in building a customized chatbot. Students can engage with numerous examples in the Jupyter notebook environment, gaining hands-on experience in prompt engineering for every topic (DeepLearning.AI 2023). Prompts mediate between users and LLMs; they affect how these models do analysis and generate data. When interacting with LLMs, users have a wide range of needs and goals. To make it more tailored to their requirements, users may direct the chat or ask for specific information using the prompts.

Research Methodology

Research context, participants, and instructional procedures. This study was conducted in the context of a mandatory "Python Programming" course, offered to Educational Technology



请帮我设计一个贪吃蛇的游戏，用python代码来实现，列出用到的各种库，分步骤进行讲解

2023/12/8 11:08:03

当然可以！下面是一个用Python编写的贪吃蛇游戏的示例代码，我将逐步解释每个步骤，并列出需要使用的库。

首先，我们需要使用 pygame 库来创建游戏窗口和处理游戏事件。确保你已经安装了 pygame 库，如果没有，请在终端中运行以下命令进行安装：

```
pip install pygame
```

接下来，我们将逐步实现贪吃蛇游戏的各个功能：

步骤1：导入所需的库

```
import pygame
import random
```

步骤2：初始化游戏

```
pygame.init()
width, height = 640, 480
screen = pygame.display.set_mode((width, height))
pygame.display.set_caption("贪吃蛇游戏")
```

Fig. 1 ChatGPT Next platform used in class.

students at a Hangzhou Normal University during Spring 2023. A quasi-experimental design was employed in this research to explore students' programming behaviors, interaction quality, and perceptions under two conditions: control (Unprompted learning condition; UL) and experimental (Prompt-based learning condition; PbL). Each condition involved 15 learners: in the control UL group, there were 9 females and 6 males, and in the experimental PbL group, 7 females and 8 males. Both groups were taught by the same teacher (the first author), who maintained a consistent teaching style across both conditions. The students had C programming experience and an agreement in written form was obtained from the review committee, confirming ethical clearance for interacting with the students and collecting research data.

Classes were delivered using the same instructional materials and followed the same teaching guidance, with the sole exception of the ChatGPT prompt application in the experimental condition. The course was segmented into three phases comprising six instructional sessions, each lasting 90 min. During Phase I and Phase II, the first four sessions covered the basic concepts of Python programming, such as Python introduction (e.g., IDLE, input(), eval(), print()), data structure (e.g., int, float, set, list, dictionary), control structure (e.g., if, for, while), functions, and methods (e.g., Recursion, Lambda). In Phase III, students were asked to develop a Python programming project aided by ChatGPT.

The structure of the instructional sessions was modeled after the Python Programming book (ICOURSE 2023). The ChatGPT Next platform, deployed by the study's first author, was utilized as a key tool to assist students' problem-solving processes (see Fig. 1). The platform was equipped with a gpt-3.5-turbo model, enabling students to initiate diverse topic conversations with ChatGPT by inputting their questions and receiving feedback. The UL group interacted with ChatGPT based on their existing knowledge (e.g., oral language), whereas the PbL group utilized additional prompts to facilitate their interactions. In Phase I and Phase II, both groups initially received lectures, oral presentations, and Python demonstrations, followed by self-guided programming tasks. In Phase III, students were tasked with developing a programming project within a 90-minute time frame.

Design of the prompt. The prompt design was tailored based on the "ChatGPT Prompt Engineering for Developers: Guidelines for Prompting" online course (DeepLearning.AI 2023). This course emphasized two essential principles for creating effective prompts (namely, crafting clear and precise instructions and providing the model with time to process) and also outlined ways to systematically engineer well-structured prompts and develop a bespoke chatbot. Various tactics were introduced under these two key principles, such as the use of delimiters, requesting structured output, and verifying satisfaction of conditions.

The rationale for adopting this course is twofold: firstly, the course was conceptualized and delivered by Andrew Ng (Founder, DeepLearning.AI) and Isa Fulford (Member of Technical Staff, OpenAI), who both have rich expertise and experience in teaching and technical domains. Secondly, this course's objective aligns well with the research context—the goal was to instruct students on how to utilize a LLM to rapidly construct robust applications, akin to the programming projects (comprising applications, websites, and so on) that students needed to develop with support from ChatGPT in our course. In line with the teaching from this course, we devised our specific prompt for the Python course, exemplified by Word Cloud project-based learning. In each instance, we offered an overall tactic explanation, supported by a detailed sample using the Word Cloud project (see Table 1). For example, for "Write clear and specific instructions" tactic, we gave students a bad example of "write me a summation function", and also a good example of "Create a summation function with Python 3 that takes two

Table 1 Examples of prompt of ChatGPT.

Prompts	Tactic	Examples
Write clear and specific instructions	Use delimiters	Distinguish code from other text, to wrap the code in triple quotes/ backticks/dashes, so that ChatGPT knows that the wrapped part is code, while the rest should just be plain text.
	Ask for structured output	Tell the model the specific output structure, not a generalized functional description, such as please list the function modules we need to use, the specific code, the comments of the code, and the result of running the code.
	Check whether conditions are satisfied	Check the assumptions required to do the task, such as please indicate whether each module is required in the code by stating [Required] or [Not Required] in front of the above functional modules.
	Few-shot prompting	Give successful examples of completing programming tasks, then ask the model to perform the task, such as how can I customize the shape of the displayed word cloud based on above successful code?
Give the model time to think	Specify the steps to complete a task	Start simple, then iterate. Don't confuse ChatGPT with complex prompts right off the bat. start with the simplest questions and gradually add complexity and edge cases. It's important to refactor and modularize as the code and complexity increase to keep your interactions and related code as small and focused as possible. Keep breaking down the code and problems into smaller chunks.
	Instruct the model to work out its own solution before rushing to a conclusion	Actively tell the model that the answer is more accurate with a little more time to think, more acid lithium, and more logical steps.

integer arguments and returns their sum", all of the examples were illustrated with pictures.

Data collection and analysis approaches. We employed three key methods to compile and scrutinize data for this study. Firstly, we documented the programming behaviors of the students using both computer screen-capture videos (mute) and platform log data. Notably, we opted to concentrate on the final session as the primary source of video data for this study (90 min per learner, totaling 2700 minutes). There were two reasons behind this choice. Initially, this session revolved around a comprehensive programming task which allowed us to monitor the complete process of students incorporating ChatGPT into problem-solving activities, thereby offering a clearer demonstration of their programming capabilities. Secondly, students in the experimental group had become more proficient at utilizing ChatGPT prompts, which resulted in increased engagement during this session, as suggested by informal observations. To analyze this data, we utilized clickstream analysis (Filva et al. 2019) to identify the programming behaviors of the students. Subsequently, an iterative coding process based on a pre-established coding framework (Sun et al. 2024b) was employed for video analysis (see Table 2).

Regardless of the video coding results, to scrutinize the behavioral patterns of the learners, we employed lag-sequential analysis (Faraone and Dorfman, 1987). This analysis included the transition frequencies between distinct behaviors and the visualized network representations across the two teaching modes. We used Yule's Q to signify the strength of transitional associations due to their utility in controlling the number of base contributions and their descriptive usefulness.

Secondly, we collected information on learners' interactions with ChatGPT (i.e., asking questions and receiving feedback) from the ChatGPT Next platform's logs. To analyze such data, we utilized Ouyang and Dai's (2022) framework for coding knowledge inquiry, allowing us to examine the quality of students' question-asking as well as their further questioning of ChatGPT's feedback. In the first step, all obtained data were systematically arranged in Excel according to the learner, performance level, and the content of the queries. Following that, inquiries posed by learners were identified and segmented into coding units (PbL: 281; UL:459), dependent on their order of appearance. In the subsequent process, two researchers individually assigned the final coding classifications, using a pre-established coding table for guidance (refer to Table 3). The consistency of these coding classifications was evaluated, achieving a reliability score of 0.81. Lastly, these data were reformatted to align with the structural requirements of cognitive network analysis. The analysis was carried out using an online tool designated for cognitive network assessment, known as webENA.

Thirdly, we focused on the learners' perceptions of ChatGPT across two classes. To gather this data, we employed surveys after the course. Inspiration for the survey design stemmed from a pre-existing model by Venkatesh and Davis (2000), which we used as a base and expanded upon using the Technology Acceptance Model (TAM) scale proposed by Sánchez and Hueros (2010). The survey was divided into two segments: the initial portion requested demographic information, past experiences, and the learners' expectations of ChatGPT, while the subsequent portion utilized a 7-point Likert scale that ranged from 1 (strongly disagree) to 7 (strongly agree). The four sub-categories considered were perceived usefulness, perceived ease of use, behavioral intention to use, and attitude towards using ChatGPT. The survey results were evaluated using an independent t-test and descriptive analysis.

Results

College students' programming behaviors in PbL and UL class. In addressing RQ1 (How did the impact of PbL on learners' programming behaviors differ from the impact of UL?), we collected and observed the student activity during the programming process. To identify any potential divergences in behavioral patterns between the two instructional paradigms, we implemented a lag-sequential analysis. Notably, the behavioral patterns of learners from both groups illuminated a blend of similarities and distinct differences.

From a frequency perspective, if we set aside irrelevant behaviors and technical issues in ChatGPT, the dominant behaviors in the PbL group were reading feedback (RF; frequency = 538), asking new questions (ANQ; frequency = 414), and understanding Python code (UPC; frequency = 249) (see Fig. 2). In contrast, the UL group applied Python coding (CP; frequency = 928) frequently, along with reading feedback (RF; frequency = 927) and asking new questions (ANQ; frequency = 414) (see Fig. 3).

Table 2 Coding framework for programming behavior.

Code	Description
Understanding task (UT)	A student transferred to the task window to understand the programming projects.
Coding in Python (CP)	A student wrote codes with the Python language in the system.
Debugging in Python (DP)	A student debugged in PyCharm.
Understanding Python codes (UPC)	A student attempted to understand the code with the mouse moving back and forth on the code.
Checking Output Window (COW)	A student checked the output window in PyCharm.
Reading console message (RCM)	A student read error messages in output console in PyCharm.
Asking new questions (ANQ)	A student asked new questions in ChatGPT.
Pasting console message (PCM)	A student pasted error messages from output console window in ChatGPT.
Pasting Python codes (PPC)	A student pasted Python codes to ChatGPT.
Reading feedback (RF)	A student read feedback in ChatGPT.
Copy and paste codes (CPC)	A student copy and paste codes from ChatGPT.
Referring to additional materials (RAM)	A student referred to additional materials from instructors.
Failure in ChatGPT (FC)	ChatGPT failed to give feedback due to technical problems (e.g., cannot connect to the server, get stuck).
Idle operation (IO)	A student had no operation.

Table 3 Encoding framework for interaction with ChatGPT.

Category	Code	Description	Sample
knowledge inquiry (KI)	Irreverent knowledge inquiry (IKI)	Students raise questions unrelated to the topic.	What historical initiative should college students possess?
	Superficial-level knowledge inquiry (SKI)	Students raise questions related to the topic, but without personal processing or perspectives	Python code for creating radar maps.
	Medium-level knowledge inquiry (MKI)	Students raise questions related to the topic, which have been processed by themselves	Create radar images using some libraries in Python and provide detailed code.
	Deep-level knowledge inquiry (DKI)	Students raise questions related to the topic, clearly articulate their viewpoints, arguments, or opinions, and provide detailed problem-solving steps	Using the Numpy and Matplotlib libraries in Python to create a Holland career analysis radar map, it is necessary to include 6 labels and provide accurate and detailed code that can run.
Feedback inquiry (FI)	Superficial-level feedback inquiry (SFI)	Student simply (dis)agreement, or simple evaluation for the feedback from ChatGPT.	Please continue writing.
	Medium-level feedback inquiry (MFI)	Students evaluate ChatGPT's feedback and ask for more simple questions.	What does linewidth=2 mean?
	Deep-level feedback inquiry (DFI)	Students evaluate the feedback on ChatGPT and provide specific examples for further information.	After entering the answer from you, there were errors such as Value Error: x and y must have the same first dimension, but have shapes (6,) and (5,). Please provide a modification plan and possible code.
Feedback from ChatGPT (FC)	General feedback from ChatGPT (GFC)	ChatGPT returned general feedback without specific problem-solving steps.	As a virtual assistant, I am able to provide you with help and support in Python programming. I don't have an actual identity, but I can help you like a programmer who is familiar with Python.
	Specific feedback from ChatGPT (SFC)	ChatGPT returned examples of text and code, and it has a clear structure or block.	Ok, taking Weibo as an example, the process of crawling resources with the theme of "postgraduate entrance examination", "English", "mathematics" and "tutorial" is roughly as follows: 1... 2...
	Correct feedback from ChatGPT (CFC)	ChatGPT returned correct feedbacks of the questions raised by students.	Ok, first you need to install Numpy and Matplotlib, which you can do with the following command: 1... 2...
	Incorrect feedback from ChatGPT (IFC)	ChatGPT returned incorrect feedback, such as incomplete answer or server error.	Ok, taking microwave as an example, the process of crawling resources with the theme of "postgraduate entrance examination", "English", "mathematics" and "tutorial" is roughly as follows: 1. Identify sites to crawl: Yes

Meanwhile, regarding sequence analyses, both the PbL and UL classes featured 38 and 41 significant programming learning sequences respectively, boasting ample interconnections across distinct codes (see Table 4). Both groups showed a propensity to read console messages (RCM) and paste console messages in ChatGPT (PCM), frequently engaging in Python code debugging (DP) and reading messages in the console (RCM) or output window (COW). More specifically, PbL students predominantly: (1) coded in Python and debugged to validate their work (CP- > DP, Yule's Q = 0.83); and (2) copied and pastes codes from ChatGPT to PyCharm and conducted debugging (CPC- > DP, Yule's Q = 0.74). In contrast, UL students mainly: (1) pasted Python codes from PyCharm to ChatGPT and inquired about new questions in ChatGPT

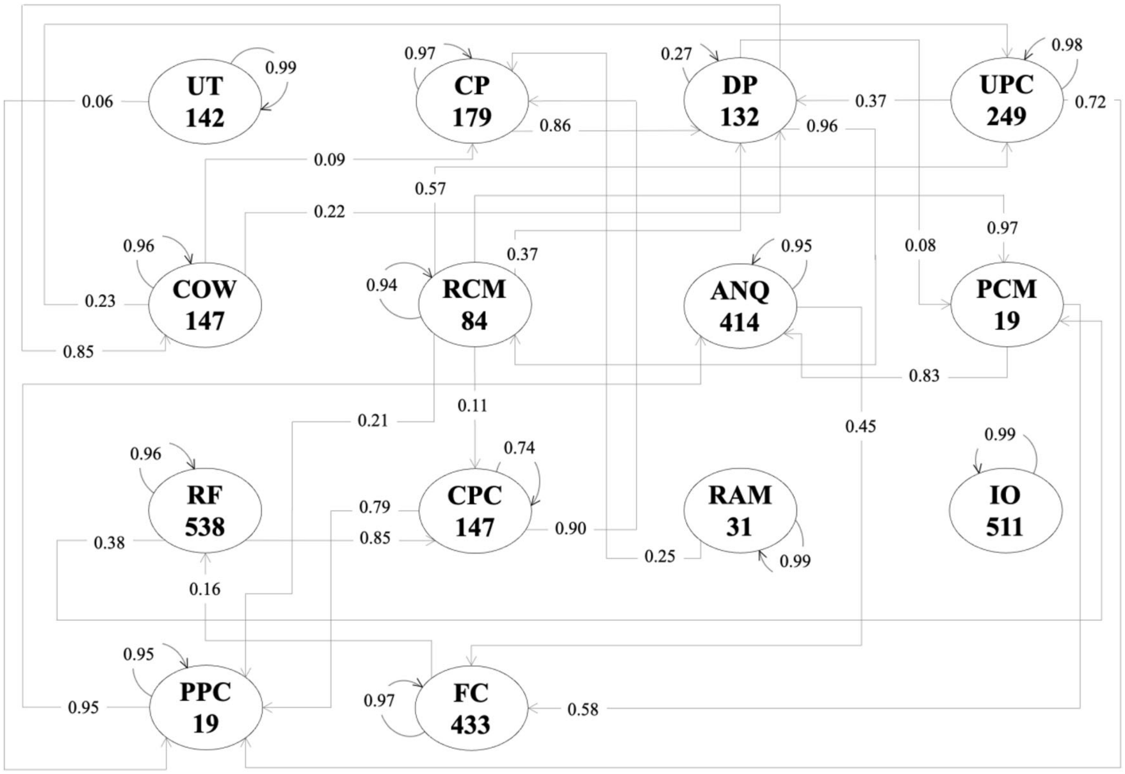


Fig. 2 Student's behavioral sequence diagram in PbL.

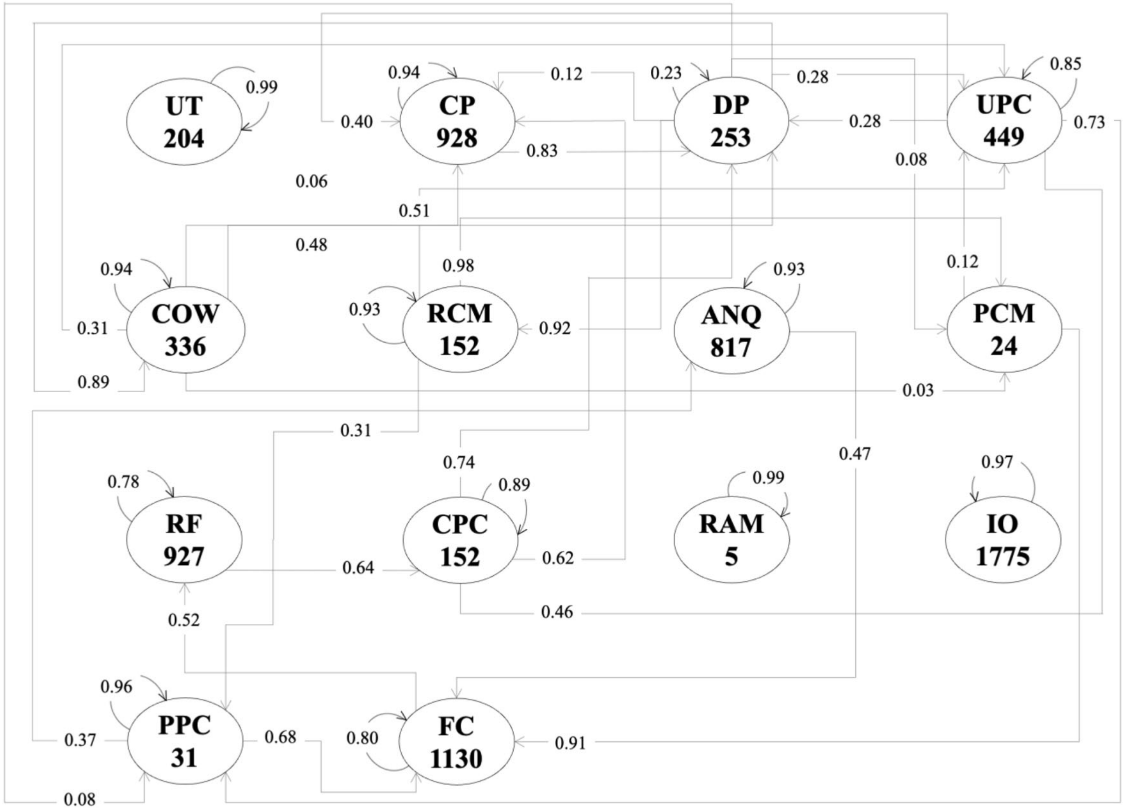


Fig. 3 Student's behavioral sequence diagram in UL.

(PPC- > ANQ, Yule's Q = 0.94) and; (2) frequently copied and adjusted codes from ChatGPT to PyCharm (CPC- > CP, Yule's Q = 0.90). These differences underscore that each class of students utilizes unique learning strategies, contingent on the contextual environment and provided resources.

College students' interaction qualities with ChatGPT in PbL and UL class. Regarding RQ2 (How did the impact of PbL on learners' interaction qualities with ChatGPT differ from the impact of UL?), we collected data on students' interaction with ChatGPT in two groups, and ENA was applied to discover differences in the cognitive level between the two groups. Fig. 4a, b exhibit the epistemic networks derived from the discourse of the PbL and UL groups. In these representations, nodes embody the codes, the lines interconnecting the nodes symbolize the corresponding connections, and line thickness is indicative of the relative co-occurrence frequency (Shaffer et al. 2016). When comparing the PbL and UL cohorts, students under the UL methodology depicted stronger connections in SKI-CFC, SKI-SFC, MFI-CFC, MFI-SFC, and SKI-GFC than those in the PbL group. On the other hand, PbL students represented a more potent MFI-CFC connection than UL students. In summary, UL students seemed more inclined to pose superficial-level questions, receive accurate responses from ChatGPT, introduce more intermediate-level questions based on the feedback from ChatGPT, and consequently receive correct responses. Conversely, students in the PbL group displayed a proclivity towards posing medium-level questions and obtaining accurate feedback from ChatGPT.

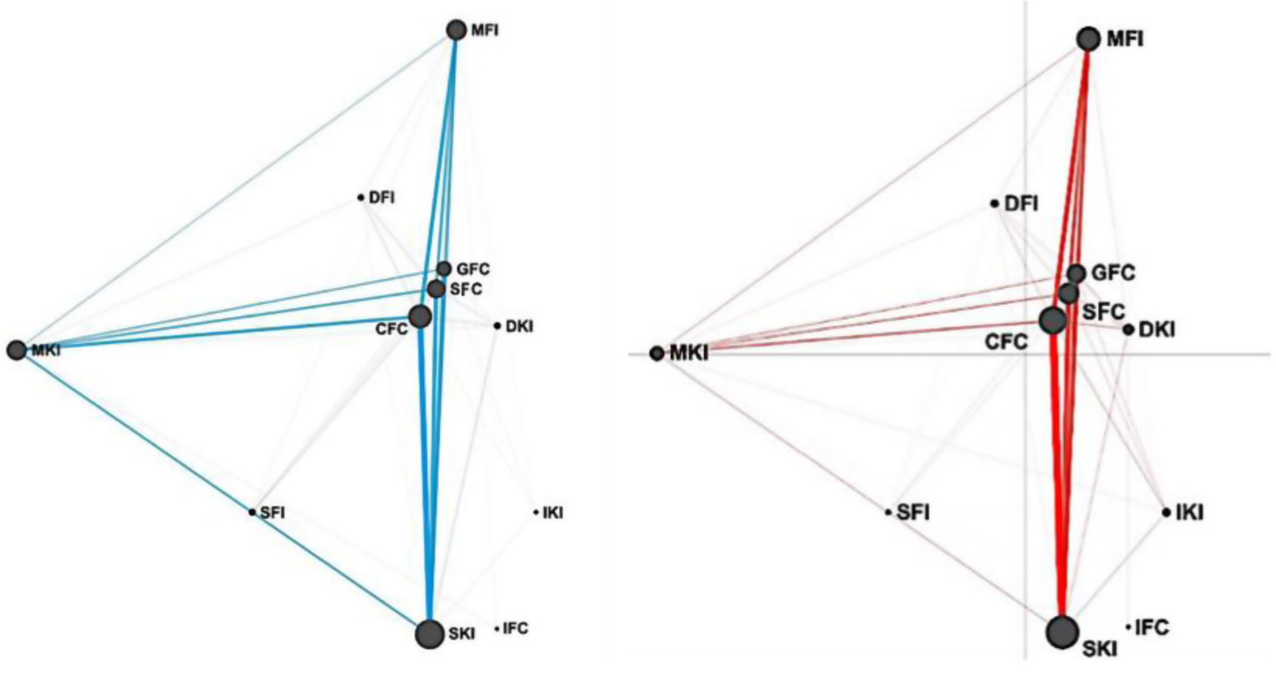
Table 4 LsA transition frequency of programming behaviors of learners in two groups.

PbL		UL	
Transition	**Yule's Q**	**Transition**	**Yule's Q**
RCM-PCM	0.98	RCM-PCM	0.97
DP-RCM	0.92	DP-RCM	0.96
DP-COW	0.89	DP-COW	0.95
CP-DP	0.83	PPC-ANQ	0.94
CPC-DP	0.74	CPC-CP	0.90

The subtracted network can represent the differences in the two clusters' category connection patterns (Sun et al. 2022). Therefore, Fig. 5 display the comparative analysis of student interactions with ChatGPT. The X-axis (first dimension) accounts for 10.7% of variable data, while the Y-axis (second dimension) accounts for an added 35.1%. To scrutinize the differences in student communications with ChatGPT between the two groups, Mann-Whitney tests were implemented. At the alpha level of 0.05, a significant differential was discovered between the two learning techniques along the X-axis (MR1). Notably, students' interactions with ChatGPT in the PbL were substantially distinct from those in the UL along the X-axis (Median FLT = 0.14, Median SLT = −0.10, U = 54.00, $p = 0.00$, $r = 0.58$), demonstrating a stark divergence in communicative patterns.

Compared to students learning in the UL group, those in the PbL group demonstrated enhanced interconnections between SKI-MKI, MKI-CFC, MKI-SFC, and MKI-GFC. Conversely, the UL students exhibited a stronger SKI-CFC link than their PbL counterparts. In essence, this suggests that PbL learners tend to generate more mid-level queries independently, based on the prompts, and receive accurate feedback from ChatGPT at a higher rate than UL learners. On the other hand, students within the UL group seem to engage in more surface-level interactions with ChatGPT, albeit still receiving correct feedback.

College students' perception towards ChatGPT in PbL and UL class. Concerning Research Question 3 (How did the impact of PbL on learners' perceptions toward ChatGPT differ from the impact of UL?), we collected student feedback through surveys and employed T-tests to examine perceptions towards ChatGPT. During the preliminary perception pre-test implemented at the



(a)

(b)

Fig. 4 Students' cognitive network of PbL (in blue) and UL (in red).

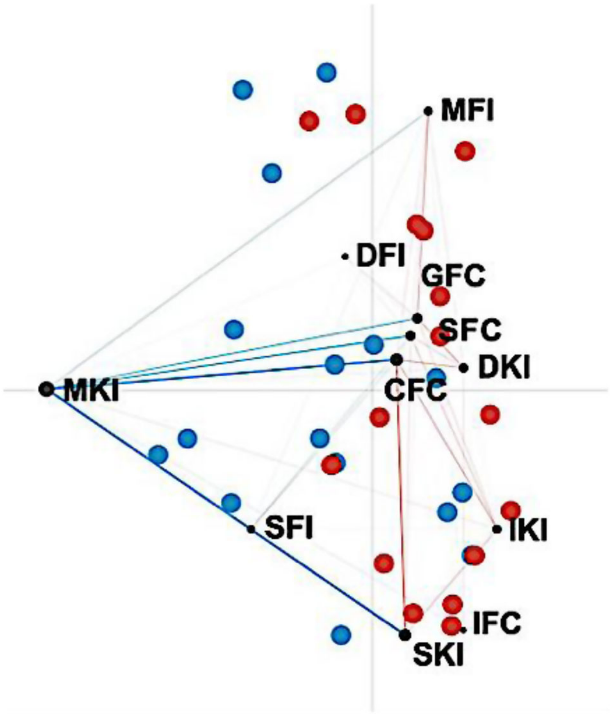


Fig. 5 Subtracted network of interaction level of PbL (in blue) and UL (in red).

Table 5 Statistic results of students' scores on the four dimensions in the post-test of perception.

	Group	*M*	SD	*t*	*p*
Perceived Usefulness	UL	4.68	1.22	−1.31	0.19
	PbL	5.23	1.06		
Perceived Ease of Use	UL	4.73	1.46	−0.27	0.78
	PbL	4.86	1.18		
Behavioral Intention to Use	UL	5.02	1.35	−1.23	0.22
	PbL	5.53	0.86		
Attitude towards Using	UL	3.82	1.28	−2.26*	0.03
	PbL	4.75	0.94		

*$p < 0.05$.

research commencement, there were no statistically significant discrepancies in perceived usefulness (t (28) = 1.05, p = 0.29), behavioral intentions to utilize (t (28) = 1.67, p = 0.11), and attitudes towards using ChatGPT (t (28) = -1.01, p = 0.32). However, a noteworthy difference was discerned in perceived ease of use (t (28) = 2.27, p = 0.03). This suggests that the UL group initially held more positive views towards the usability of ChatGPT. After the intervention (see Table 5), learners in the PbL group reported higher average scores in perceived usefulness, ease of use, behavioral intention to utilize, and attitudes toward using ChatGPT. T-tests revealed a statistically significant difference favoring PbL, specifically about the attitude towards using ChatGPT (t (28) = -2.26, p = 0.03). In essence, these results indicate that, with the assistance of ChatGPT prompts, learners from the PbL cohort demonstrate significantly improved attitudes towards ChatGPT usage and perceive it to be easier to use than their UL counterparts in post-intervention.

In examining the influence of different prompts on learning, students found the guidance to seek structured output most useful, as it allowed clearly organized solutions, enhancing understanding of problem-solving steps. The use of delimiters was also appreciated, helping manage input and output, making learning more efficient with ChatGPT. Finally, students valued instructions that promoted thoughtful problem-solving by the model, offering insight into its logic, encouraging methodical approaches in learners, and facilitating proficiency in programming and analysis (see Fig. 6).

Discussion

The primary objective of this research was to explore the impact of prompts while using ChatGPT programming learning on college students' programming behaviors, interaction quality, and perceptions by comparing a class utilizing prompt-based learning with a class directed under conditions without prompts. The findings present an interesting array of shared and unique characteristics in the programming behaviors of learners in both groups, with notable differences primarily attributed to the existence or absence of prompts.

Firstly, the deployment of lag-sequential analysis allowed for an examination of the students' activities during the programming process, revealing distinct patterns of behavior reflective of the two learning paradigms. In terms of frequency, PbL students exhibited a strong engagement with feedback, a high incidence of question-asking, and a notable effort to understand Python code. This pattern aligns with the adaptability of ChatGPT, which enables individuals to pursue self-directed learning, surpassing the boundaries of traditional educational settings (Gayed et al. 2022). The PbL approach appears to cultivate a reflective and inquisitive mindset, echoing the findings of Yilmaz and Karaoglan Yilmaz (2023a), who found that PbL enhances self-regulated learning. On the other hand, UL students demonstrated a greater frequency of applying Python coding, which suggests a more hands-on, exploratory approach. But the fact that UL students copy and change code from ChatGPT so often might mean that they are too dependent on outside help, which could stop them from developing deep problem-solving skills (Krupp et al. 2023). The sequence analysis further differentiates the learning behaviors between PbL and UL. PbL students displayed sequences that suggest an iterative process of coding and debugging, which is indicative of a deeper engagement with the material (Sun et al. 2024c). In contrast, UL students' sequences reveal a pattern of seeking immediate solutions through copying and adjusting codes, which might reflect a more surface-level engagement with the task at hand (Mayer 2004).

Secondly, the ENA findings suggest that the PbL approach fosters deeper cognitive engagement, as evidenced by the stronger connections between mid-level questioning and correct feedback from ChatGPT. This aligns with the constructivist theories, which posit that learners benefit from engaging in tasks that require them to apply and extend their existing knowledge and skills. The PbL method, with its emphasis on problem-solving, appears to support this type of cognitive development (Ekin 2023; McTear et al. 2023). The stronger connections in the UL group's surface-level interactions, shown by the SKI-CFC link, often lead to cognitive processing at the surface level. In contrast, our study demonstrates that PbL environments promote an interaction dynamic that encourages learners to engage in more meaningful and complex discourse with ChatGPT, thereby potentially enhancing their understanding and retention of the material. The subtracted network and the significant differences found in the comparative analysis show that the PbL and UL groups communicate in very different ways. This finding is consistent with the work of Sun et al. (2022), who found that the structure of learning activities significantly influences the quality of student interactions with learning technologies. Moreover, the enhanced interconnections between mid-level knowledge indicators in the

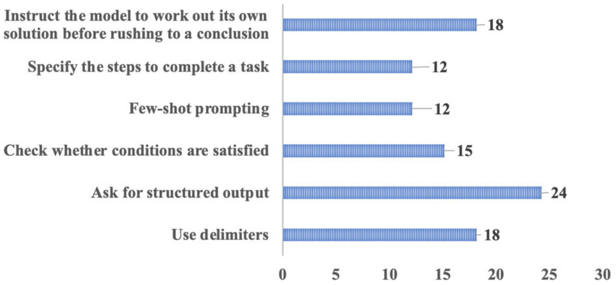


Fig. 6 Ranking of students' preferred prompts of ChatGPT. The Y-axis represents the different prompts provided for students, and the X-axis represents the percentage of students who chose the corresponding prompt.

PbL group suggest that this approach facilitates a more integrated understanding of concepts, as learners are more frequently connecting different levels of knowledge. This observation supports the assertion by Shaffer et al. (2016) that the structure of knowledge connections within a learning environment can impact the depth of student understanding and learning outcomes.

Thirdly, PbL learners reported a more favorable perception of ChatGPT across all measured constructs, which posits that users' continued intention to use technology is influenced by their expectations, perceived performance, and satisfaction. The structured prompts in PbL likely helped to align learners' expectations with the actual performance of ChatGPT, leading to higher satisfaction and a more positive attitude. Perceived usefulness and ease of use are critical factors in the acceptance and adoption of new technologies (Davis 1989). The PbL approach, with its structured prompts, seems to have provided a scaffold that enhanced the learners' understanding and efficiency in using ChatGPT. This scaffolding aligns with McTear et al. (2023), who found that learners can achieve higher levels of understanding with the support of more knowledgeable others or tools. Moreover, the preference for prompts that encourage structured outputs and the use of delimiters indicate that learners value clarity and organization in the learning process, emphasizing the importance of a coherent presentation of information in enhancing comprehension and learning. The appreciation for prompts that encourage thoughtful problem-solving by the model suggests that learners are seeking not just answers but also understanding of the underlying logic, where learning is an active, constructive process.

In conclusion, our study suggests that the effectiveness of AI tools like ChatGPT in educational settings is significantly influenced by the pedagogical context in which they are employed. The PbL approach appears to amplify the benefits of ChatGPT by aligning with constructivist principles and fostering positive attitudes towards the use of ChatGPT in learning. These findings have implications for the design of educational interventions and the integration of ChatGPT in the classroom, emphasizing the need for a pedagogically sound approach to maximize the potential of these advanced technologies.

Implications

Considering the research findings, this study proposes pedagogical implications intended to guide educators in the effective integration of AI tools like ChatGPT into their teaching practices, especially in the context of programming education. To begin with, educators are advised to integrate structured prompts into their pedagogical strategies when employing these advanced tools to enhance learning outcomes. Structured prompts are instrumental in guiding students through the learning process, prompting them to actively engage with the material, which is critical for deeper comprehension and retention (Yilmaz and Karaoglan Yilmaz 2023a). The design of these tasks should be such that they necessitate student interaction with feedback, incentivize question-asking, and encourage reflective thinking about their understanding (Babe et al. 2023). Moreover, the act of asking questions is a fundamental component of the learning process, as it drives learners to seek clarification and deepen their understanding (McTear et al. 2023). The integration of structured prompts into the use of AI tools in programming education is a pedagogical strategy that can lead to more meaningful learning experiences. For example, instructors have the potential to deepen students' comprehension of the material through the provision of comprehensive explanations and examples that demonstrate the use of ChatGPT in programming. In the realm of AI technology, a "prompt" sets the stage for a task or discourse that the language model endeavors to complete. The crafting of a suitable prompt takes on critical importance, as the model aims to produce text that fits seamlessly with the context defined by the initial prompt, aligning with the findings of Liu et al. (2023).

In addition, since the cultivation of problem-solving skills is a cornerstone of effective programming education, instructors should share their way of utilizing feedback from ChatGPT. To leverage ChatGPT's feedback effectively in programming education, instructors should guide students on how to interpret the feedback critically and integrate the feedback into their work effectively. This could involve understanding the suggestions provided, identifying the root causes of issues, recognizing patterns in the feedback that could indicate deeper conceptual misunderstandings (Gayed et al. 2022), and not just making direct changes but understanding why those changes are necessary. Instructors can also model this process by working through examples and demonstrating how to use feedback to refine and improve code. Additionally, students should be encouraged to reflect on the feedback they receive and consider its application to the problem they are attempting to solve. Prompts can facilitate reflection by asking students to summarize what they learned from the feedback and how they might apply this learning to similar problems in the future (Gayed et al. 2022).

Moreover, instructors also need to actively discourage negative practices such as code copying without any further consideration. While this may offer short-term convenience, as Krupp et al. (2023) suggest, it hinders the development of vital problem-solving skills and a deep understanding of programming. Such practices lead to a superficial engagement with the material, where students may complete tasks without truly grasping the principles or being able to apply knowledge in new contexts. By focusing on these aspects, instructors can foster a learning environment where feedback from ChatGPT is not just a tool for answers but a catalyst for developing robust programming expertise and problem-solving acumen.

Conclusion, limitations, and future directions

ChatGPT, an advanced AI language model, has demonstrated its ability to support computer programming tasks by offering guidance, responding to queries, and producing code snippets; harnessing the power of prompting is a technique that can guide the discourse with such a large language model, driving it to disclose pertinent information. Our quasi-experimental study assesses the impact of prompt-driven interaction on college students' programming behaviors, the quality of their interactions, and their perceptions towards ChatGPT. The study juxtaposes the outcomes of a prompt-based learning approach against those of a control group engaging in unprompted learning scenarios. The findings underscore the significant role that prompt structuring plays in the domain of programming education. Nevertheless, the study's scope presents certain limitations that warrant consideration. Firstly, the generalizability of the results is potentially constrained by the singular prompt version employed; subsequent research should endeavor to validate these findings across a diverse array of prompt variations, and explore the use of structured prompts to provide students with clear guidance when engaging with Chatbot. Secondly, the pivotal role of instructional design in the programming learning trajectory suggests that further research should investigate the integration of prompt-based methodologies at various instructional stages. In particular, future studies may explore the effectiveness of using prompt-based methods to encourage students to think about what they have learned, considering the feedback they receive and how it can be uptake and applied to problem-solving. We will also examine how such prompts can help develop critical thinking skills in students. By investigating these issues, we may be able to fine-tune the use of Chatbot in programming education and support better learning outcomes among students. Lastly, the six-week duration of the study highlights the necessity for more extensive longitudinal research to better understand student perceptions of prompt utilization over extended periods. Such research would offer a more comprehensive view of the long-term effects of prompts on learning outcomes in programming education.

The present study's findings suggest that the integration of structured prompts within programming instruction can improve the learning experience when interacting with ChatGPT. While acknowledging the limitations encountered in this research, it is imperative to pursue further academic investigation to expand upon these initial findings and fully ascertain the capabilities of AI language models in optimizing programming education. Consequently, it can be reasonably posited that the utilization of tailored prompts in ChatGPT may serve as a potent pedagogical instrument, benefiting educators and learners by enhancing the instructional experience in the realm of computer programming.

Data availability

Data will be made available on reasonable request.

Received: 30 January 2024; Accepted: 24 October 2024;
Published online: 31 October 2024

References

Ahmed MA (2023) ChatGPT and the EFL classroom: Supplement or substitute in Saudi Arabia's eastern region. Inf Sci Lett 12(7):2727–2734. https://doi.org/10.18576/isl/120704

Andersen R, Mørch AI, Litherland KT (2022) Collaborative learning with block-based programming: investigating human-centered artificial intelligence in education. Behav Inf Technol 41(9):1830–1847. https://doi.org/10.1080/0144929x.2022.2083981

Avila-Chauvet L, Mejía D, & Acosta Quiroz CO (2023) Chatgpt as a support tool for online behavioral task programming. Retrieved 20 Feburary, 2024, from https://elkssl50f6c6d9a165826811b16ce3d7452cf8elksslauthserver.casb.hznu.edu.cn/abstract=4329020

Babe HM, Nguyen S, Zi Y, Guha A, Feldman MQ, Anderson CJ (2023) StudentEval: A benchmark of student-written prompts for large language models of code. Cornell University Library, arXiv.org, Ithaca, https://doi.org/10.48550/arxiv.2306.04556

Balanskat A, Engelhardt K (2015) Computing our future. Computer programming and coding – Priorities, school curricula and initiatives across Europe–update 2015. European Schoolnet, Brussels

Biswas S (2022) Role of ChatGPT in Computer Programming. Mesopotamian J Comput Sci 8–16. https://doi.org/10.58496/mjcsc/2022/004

Brown T, Mann B, Ryder N, Subbiah M, Kaplan J. D, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, & Amodei D (2020). Language models are few-shot learners. In H Larochelle, M Ranzato, R Hadsell, MF Balcan, and H Lin eds. Advances in NeurIPS, (v 33), (pp. 1877–1901). Curran Associates, Inc., URL https://elksslc6aca02ee3bce85ceac323fb2b0abf87elksslauthserver.casb.hznu.edu.cn/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

Cooper G (2023) Examining science education in ChatGPT: An exploratory study of generative artificial intelligence. J Sci Educ Technol 32(3):444–452

Davis FD (1989) Perceived usefulness, perceived ease of use and user acceptance of information technology. MIS Quartely 13(3):319–340

DeepLearning.AI. (2023) Retrieved 20 Feburary, 2024, from https://elksslc767dbb71ea7c8497a1d1906bf794435elksslauthserver.casb.hznu.edu.cn/short-courses/chatgpt-prompt-engineering-for-developers/

Dengler K, Matthes B (2018) The impacts of digital transformation on the labour market: Substitution potentials of occupations in Germany. Technol Forecast Soc Change 137:304–316. https://doi.org/10.1016/j.techfore.2018.09.024

Denny P, Leinonen J, Prather J, Luxton-Reilly A, Amarouche T, Becker BA, Reeves BN (2023) Promptly: Using prompt problems to teach learners how to effectively utilize AI code generators. Cornell University Library, arXiv.org, Ithaca, 10.48550/arxiv.2307.16364

Dwivedi YK, Kshetri N, Hughes L, Slade EL, Jeyaraj A, Kar AK, Baabdullah AM, Koohang A, Raghavan V, Ahuja M, Albanna H, Albashrawi MA, Al-Busaidi AS, Balakrishnan J, Barlette Y, Basu S, Bose I, Brooks L, Buhalis D, Wright R (2023) So what if ChatGPT wrote it?" multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. Int J Inf Manag 71:102642. https://doi.org/10.1016/j.ijinfomgt.2023.102642

Eason C, Huang R, Han-Shin C, Tseng Y, Liang-Yi L (2023) GPTutor: A ChatGPT-powered programming tool for code explanation. Cornell University Library, arXiv.org, Ithaca, 10.48550/arxiv.2305.01863

Ekin S (2023) Prompt engineering for ChatGPT: A quick guide to techniques, tips, and best practices. Retrieved Jan 10, 2024 from https://doi.org/10.36227/techrxiv.22683919

Faraone SV, Dorfman DD (1987) Lag sequential analysis: Robust statistical methods. Psychol Bull 101(2):312–323. https://doi.org/10.1037/0033-2909.101.2.312

Filva DA, Forment MA, Garcıa-Penalvo FJ, Escudero DF, Casan MJ (2019) Clickstream for learning analytics to assess students' behavior with scratch. Future Gener Computer Syst 93:673–686. https://doi.org/10.1016/j.future.2018.10.057

Gayed JM, Carlon MKJ, Oriola AM, Cross JS (2022) Exploring an AI-based writing assistant's impact on English language learners. Computers Educ: Artif Intell 3:100055. https://doi.org/10.1016/j.caeai.2022.100055

ICOURSE (2023) retrieved from 28 July, 2023 from https://elkssl0e848bfdddb2460a97621c537a9f3f13elksslauthserver.casb.hznu.edu.cn/course/BIT-268001?from=searchPage&outVendor=zw_mooc_pcssjg_

Jeon J, Lee S (2023) Large language models in education: A focus on the complementary relationship between human teachers and ChatGPT. Edu Info Technol 1–20. https://doi.org/10.1007/s10639-023-11834-1

Kashefi A, Mukerji T (2023) ChatGPT for programming numerical methods. J Mach Learn Modeling Comput 4(2):1–74. https://doi.org/10.1615/jmachlearnmodelcomput.2023048492

Kohnke L, Moorhouse BL, Zou D (2023) ChatGPT for language teaching and learning. RELC J 54(2):537–550. https://doi.org/10.1177/00336882231162868

Krupp L, Steinert S, Kiefer-Emmanouilidis M, Avila KE, Lukowicz P, Kuhn J, Küchemann S, Karolus J (2023) Unreflected acceptance – investigating the negative consequences of ChatGPT-assisted problem solving in physics education. Cornell University Library, arXiv.org, Ithaca, 10.48550/arxiv.2309.03087

Liu J, Li Q, Sun X, Zhu Z, Xu Y (2021) Factors influencing programming self-efficacy: an empirical study in the context of Mainland China. Asia Pac J Educ 43(3):835–849. https://doi.org/10.1080/02188791.2021.1985430

Liu P, Yuan W, Fu J, Jiang Z, Hayashi H, Neubig G (2023) Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Comput Surv 55(9):1–35

Liu G, Ma C (2023) Measuring EFL learners' use of ChatGPT in informal digital learning of English based on the technology acceptance mode, Innov Lang Learn Teach (online), 1–19. https://doi.org/10.1080/17501229.2023.2240316

Liu I-F, Hung H-C, Liang C-T (2023) A Study of Programming Learning Perceptions and Effectiveness under a Blended Learning Model with Live Streaming: Comparisons between Full-Time and Working Students. Interactive Learn Environ (online), 1–15, https://doi.org/10.1080/10494820.2023.2198586

Lu OHT, Huang JCH, Huang AYQ, Yang SJH (2017) Applying learning analytics for improving students engagement and learning outcomes in an MOOCs enabled collaborative programming course. Interact Learn Environ 25(2):220–234. https://doi.org/10.1080/10494820.2016.1278391

Mayer RE (2004) Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instruction. Am Psychologist 59(1):14–19

McTear M, Varghese Marokkie S, Bi Y (2023) A comparative study of chatbot response generation: Traditional approaches versus large language models.In: Jin Z, Jiang Y, Buchmann RA, Bi Y, Ghiran AM, Ma W (eds) Lecture Notes in

Computer Science. Knowledge Science, Engineering and Management. KSEM 2023. Springer, Cham 14118. https://doi.org/10.1007/978-3-031-40286-9_7

Mollick E, Mollick L (2023) Assigning AI: Seven approaches for students, with prompts. Cornell University Library, arXiv.org, Ithaca, 10.48550/arxiv.2306.10052

Nouri J, Zhang L, Mannila L, Noren E (2020) Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. Educ Inq 11(1):1–17

Ofosu-Ampong K, Acheampong B, Kevor M-O, Amankwah-Sarfo F (2023) Acceptance of artificial intelligence (ChatGPT-4) in education: Trust, innovativeness and psychological need of students. Info Knowledge Manag 13(4). https://doi.org/10.7176/ikm/13-4-03

Ou Q, Liang W, He Z, Liu X, Yang R, Wu X (2023) Investigation and analysis of the current situation of programming education in primary and secondary schools. Heliyon 9(4):e15530. https://doi.org/10.1016/j.heliyon.2023.e15530

Ouh EL, Benjamin KSG, Shim KJ, Wlodkowski S (2023) ChatGPT, can you generate solutions for my coding exercises? an evaluation on its effectiveness in an undergraduate java programming course. Cornell University Library, arXiv.org, Ithaca, https://doi.org/10.48550/arxiv.2305.13680

Ouyang F, Dai X (2022) Using a three-layered social-cognitive network analysis framework for understanding online collaborative discussions. Australas J Educ Technol 38(1):164–181. https://doi.org/10.14742/ajet.7166

Ouyang L, Wu J, Xu J, Almeida D, Wainwright CL, Mishkin P, Zhang C, Agarwal S, Slama K, Ray A, Schulman J, Hilton J, Kelton F, Miller L, Simens M, Askell A, Welinder P, Christiano P, Leike J, Lowe R (2022) Training language models to follow instructions with human feedback. Cornell University Library, arXiv.org, Ithaca, https://doi.org/10.48550/arxiv.2203.02155

Phung T, Pădurean V, Cambronero J, Gulwani S, Kohn T, Majumdar R, Singla A, Soares G (2023) Generative AI for programming education: Benchmarking ChatGPT, GPT-4, and human tutors. Cornell University Library, arXiv.org, Ithaca, https://doi.org/10.48550/arxiv.2306.17156

Pryzant R, Iter D, Li J, Lee YT, Zhu C, Zeng M (2023) Automatic prompt optimization with "gradient descent" and beam search. Cornell University Library, arXiv.org, Ithaca, https://doi.org/10.48550/arxiv.2305.03495

Rawas S (2023) ChatGPT: Empowering lifelong learning in the digital age of higher education. Educ Info Technol (online), 1–14. https://doi.org/10.1007/s10639-023-12114-8

Sánchez RA, Hueros AD (2010) Motivational factors that influence the acceptance of Moodle using TAM. Computers Hum Behav 26(6):1632–1640. https://doi.org/10.1016/j.chb.2010.06.011

Schick T, Schütze H (2021) Exploiting cloze questions for few shot text classification and natural language inference. Cornell University Library, arXiv.org, Ithaca, https://doi.org/10.48550/arxiv.2001.07676

Shaffer DW, Collier W, Ruis AR (2016) A tutorial on epistemic network analysis: analyzing the structure of connections in cognitive, social, and interaction data. J Learn Analytics 3(3):9–45

Strzelecki A (2023) To use or not to use ChatGPT in higher education? A study of students' acceptance and use of technology. *Interactive Learning Environments, (online)*, 1–14. https://doi.org/10.1080/10494820.2023.2209881

Sun D, Boudouaia A, Zhu C et al. (2024a) Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study Int J Edu Technol Higher Educ 21:14. https://doi.org/10.1186/s41239-024-00446-5

Sun D, Ouyang F, Li Y, Zhu C (2021b) Comparing learners' knowledge, behaviors, and attitudes between two instructional modes of computer programming in secondary education. Int J STEM Educ 8(1):54–54. https://doi.org/10.1186/s40594-021-00311-1

Sun L, Guo Z, Zhou D (2022) Developing K-12 students' programming ability: A systematic literature review. Educ Inf Technol 27:7059–7097. https://doi.org/10.1007/s10639-022-10891-2

Sun M, Wang M, Wegerif R, Jun P (2022) How do students generate ideas together in scientific creativity tasks through computer-based mind mapping? Computers Educ 176:104359. https://doi.org/10.1016/j.compedu.2021.104359

Sun D, Looi C, Li Y, Zhu C, Zhu C, Cheng M (2024b) Block-based versus text-based programming: A comparison of learners' programming behaviors, computational thinking skills and attitudes toward programming. Educ Technol Res Develop, online. https://doi.org/10.1007/s11423-023-10328-8

Sun D, Ouyang F, Li Y, Zhu C, Zhou Y (2024c) Using multimodal learning analytics to understand effects of block-based and text-based modalities on computer programming. J Comput Assisted Learn, (online), 1 Int J Edu Technol Higher Edu14. https://doi.org/10.1111/jcal.12939

Sun L, Hu L, Zhou D (2021a) Which way of design programming activities is more effective to promote K-12 students' computational thinking skills? A meta-analysis. J Comput Assist Learn (online), 1Int J Edu Technol Higher Edu 15. https://doi.org/10.1111/jcal.12545

Surameery NMS, Shakor MY (2023) Use chatgpt to solve programming bugs. Int J Inf Technol Computer Eng 3(1):17–22

Tam A (2023) What are large language models. Retrieved Jan 10, 2024 from https://elkssl6ed619682db9e40376a67ef6f6364a3belksslauthserver.casb.hznu.edu.cn/what-are-large-languagemodels/#:~:text=There%20are%20multiple%20large%20language,language%20and%20can%20generate%20text

Thongkoo, K Daungcharone, J & Thanyaphongphat. (2020). Students' acceptance of digital learning tools in programming education course using technology acceptance model. In Proceedings of Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (pp. 377–380)

Tian H, Lu W, Li TO, Tang X, Cheung S, Klein J, Bissyandé TF (2023) Is ChatGPT the ultimate programming assistant – how far is it? Cornell University Library, arXiv.org, Ithaca, https://doi.org/10.48550/arxiv.2304.11938

Venkatesh V, Davis FD (2000) A theoretical extension of the technology acceptance model: Four longitudinal field studies. Manag Sci 46(2):186–204. https://doi.org/10.1287/mnsc.46.2.186.11926

Vukojičić, M, & Krstić, J (2023) ChatGPT in programming education: ChatGPT as a programming assistant. *Journal for Contemporary Education, (online)*, 7-13

Wei J, Bosma M, Zhao VY, Guu K, Adams WY, Lester B, Du N, Dai AM, Le QV (2021) Finetuned language models are zero-shot learners. Cornell University Library, arXiv.org, Ithaca, https://doi.org/10.48550/arxiv.2109.01652

White J, Fu Q, Hays S, Sandborn M, Olea C, Gilbert H, Elnashar A, Spencer-Smith J, Schmidt DC (2023) A prompt pattern catalog to enhance prompt engineering with ChatGPT. Cornell University Library, arXiv.org, Ithaca, https://doi.org/10.48550/arxiv.2302.11382

Xie Y, Boudouaia A, Xu J, AL-Qadri AH, Khattala A, Li Y, Aung YM (2023) A study on teachers' continuance intention to use technology in English instruction in western China junior secondary schools. Sustainability 15(5):4307. https://doi.org/10.3390/su15054307

Xu L, Chen Y, Cui G, Gao H, Liu Z (2022) Exploring the universal vulnerability of prompt-based learning paradigm. Cornell University Library, arXiv.org, Ithaca, https://doi.org/10.48550/arxiv.2204.05239

Yi YJ, You S, Bae BJ (2016) The influence of smartphones on academic performance. Libr Hi Tech 34(3):480–499. https://doi.org/10.1108/lht-04-2016-0038

Yilmaz R, Karaoglan Yilmaz FG (2023b) The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. Computers Educ Artif Intell 4:100147. https://doi.org/10.1016/j.caeai.2023.100147

Yilmaz R, Karaoglan Yilmaz FG (2023a) Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. Computers Hum Behav: Artif Hum 1(2):100005. https://doi.org/10.1016/j.chbah.2023.100005

Yin, W, Hay, J, and Roth, D (2019) Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*, in: Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3912–3921. https://doi.org/10.18653/v1/D19-1404

Acknowledgements

The authors extend their heartfelt appreciation to all the individuals involved, acknowledging their enthusiasm and collaboration throughout this study.

Author contributions

Dan Sun designed and performed the study. Dan Sun, Azzeddine Boudouaia, Jie Xu analysed and interpreted the data. Dan Sun and Azzeddine Boudouaia wrote the initial draft of the paper. Junfeng Yang critically revised the manuscript. All authors have contributed to, read, and approved the final version of the manuscript.

Funding

This project received financial support from the China National Social Science Key Project, Research on the Ethics and Limits of AI in Education Scenarios (ACA220027), National Natural Science Foundation of China [Grant No. 62307011], and ethical risks and precautionary measures for new generation AI in education, Zhejiang social science key projects (24QNYC14ZD).

Competing Interests

The authors declare no competing interests.

Ethical approval

This research project has been reviewed and approved by the Ethical Review Committee at Jinghengyi School of Education, Hangzhou Normal University on June 29, 2023. Our research team confirms that all procedures involving human participants were duly performed in rigorous accordance with the ethical standards stipulated by the Declaration of Helsinki. This includes, but is not limited to, obtaining informed consent from all human participants, ensuring their privacy and confidentiality, and minimizing any potential harm or discomfort. The approval ID for this research project is 2023017. The scope of approval covers all aspects of the research, including but not limited to participant recruitment, data collection, analysis, and dissemination of results. Any amendments or additional protocols related to the research have been reviewed and approved by the Ethical Review Committee at Jinghengyi School of Education, Hangzhou Normal University under the same approval number or a new one issued accordingly.

Informed consent

In compliance with ethical guidelines, instructor (the first author of this study) collected consent form from all the participants, which stating the details of the study's objectives, their involvement, potential risks and benefits, and assurances of confidentiality and anonymity. By signing the consent form, participants were giving their consent for the following: participation in the study, which may include surveys, computer screen-capture videos, and platform log data collection; use of the data collected for research purposes, which may include analysis and publication; publication of the results of the study, with no personal identifiers used to maintain confidentiality; all information collected will be kept confidential, data will be stored securely and will only be accessible to the research team, participants' identity will not be disclosed in any reports or publications resulting from the study. No vulnerable individuals and payment or incentives were involved in this study.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1057/s41599-024-03991-6.

Correspondence and requests for materials should be addressed to Azzeddine Boudouaia.