

Scaffolding Computational Thinking With ChatGPT

Jian Liao , Linrong Zhong , Longting Zhe , Handan Xu , Ming Liu , and Tao Xie 

Abstract—ChatGPT has received considerable attention in education, particularly in programming education because of its capabilities in automated code generation and program repairing and scoring. However, few empirical studies have investigated the use of ChatGPT to customize a learning system for scaffolding students' computational thinking. Therefore, this article proposes an intelligent programming scaffolding system using ChatGPT following the theoretical framework of computational thinking and scaffolding. A mixed-method study was conducted to investigate the affordance of the scaffolding system using ChatGPT, and the findings show that most students had positive attitudes about the proposed system, and it was effective in improving their computational thinking generally but not their problem-solving skills. Therefore, more scaffolding strategies are discussed with the aim of improving student computational thinking, especially regarding problem-solving skills. The findings of this study are expected to guide future designs of generative artificial intelligence tools embedded in intelligent learning systems to foster students' computational thinking and programming learning.

Index Terms—Artificial-intelligence-generated content (AIGC), ChatGPT, computational thinking (CT), scaffolding.

I. INTRODUCTION

LAUNCHED by OpenAI in November 2022, ChatGPT has gained many users in a short period of time, and it has received significant attention in education because it is a major breakthrough in the field of artificial intelligence (AI), providing promising benefits for learning and teaching. Researchers and practitioners from various disciplines now use ChatGPT for educational applications, such as manuscript writing, assisted teaching, and computer education [1], [2], [3]. ChatGPT changes the way we learn in the digital age by enabling more natural and

intuitive forms of communication with flexible and accessible AI [4], [5].

As an important skill in the 21st century [6], [7], computational thinking (CT) involves using fundamental concepts from computer science to solve problems; it is closely linked to programming skills, and programming is regarded as the most appropriate way to enhance students' CT [8], [9], [10]. Although ChatGPT is used widely in various fields, its application in programming education to foster CT is still in the early exploratory stage.

Moreover, scholars have claimed that ChatGPT might introduce new challenges and concerns in education [11], [12]. For instance, students may become overly dependent on ChatGPT, as excessive reliance on AI tools could impede students' efforts to actively pursue their learning goals through independent thinking [13].

The present study addresses these issues by using scaffolding as a theoretical lens to frame the application of ChatGPT in programming learning to promote students' active learning. Two general design principles were proposed for applying scaffolding in programming course to improve students' CT with the support of ChatGPT: 1) integrating ChatGPT with the key factors of scaffolding and 2) designing interactive modules with ChatGPT following the developmental process of CT.

Consequently, an intelligent programming scaffolding system using ChatGPT (IPSSC) was developed for a Data Structures and Algorithms course at university level. Instead of chatting with ChatGPT directly, three modules called solution assessment (SA), code assessment (CA), and free interaction (FI) were designed following a modified CT framework to assist in scaffolding students to finish the coding tasks of the course. Therefore, this study designed three interactive modules in the IPSSC according to the five aspects of the CT framework proposed by Shute et al. [14], i.e., decomposition, abstraction, algorithms, debugging, and iterations, the aim being to cultivate various aspects of students' CT.

Accordingly, a mixed-method study was conducted, with the CT Scale (CTS) [7] used as a pre- and posttest instrument in a quasi-experimental study to evaluate students' CT levels. A questionnaire was used to collect students' perceived effects and attitudes toward the proposed system. Moreover, nine participants were purposely selected for interviews based on the number of interactions between students and the IPSSC to triangulate students' experience on the proposed system. In addition, lag sequential analysis and clustering analysis were used in this study to mine the interaction patterns because the IPSSC can record students' interactions with ChatGPT. The findings of this study are expected to be used to explore new teaching

Manuscript received 31 August 2023; revised 16 January 2024 and 5 April 2024; accepted 9 April 2024. Date of publication 24 April 2024; date of current version 30 May 2024. This work was supported in part by the Research Projects Funded by the National Natural Science Foundation of China under Grant 62177039, in part by the Key Projects of Scientific Research in Higher Education in Chongqing under Grant cqj23005B, in part by the Research Projects Funded by the National Natural Science Foundation of China under Grant 62277043, in part by Chongqing Natural Science Foundation Innovation and Development Joint Fund Project under Grant CSTB2023NSCQ-LZX0094, in part by the Research Funds for the Southwest University under Grant SWU2108023, and in part by the Innovation Research 2035 Pilot Plan of Southwest University under Grant SWUPilotPlan002. (Corresponding author: Ming Liu.)

This work involved human subjects in its research. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Review Board at Southwest University under Application No. SWUFEIRB 2023-0201, and performed in line with university requirements.

The authors are with the Faculty of Education, Southwest University, Chongqing 400715, China (e-mail: leojames@swu.edu.cn; linrongzhong032@163.com; longtingzhe@163.com; xuhandan226@163.com; mingliu@swu.edu.cn; xietao@swu.edu.cn).

Digital Object Identifier 10.1109/TLT.2024.3392896

strategies with the support of AI tools based on ChatGPT and provide design principles for the design and implementation of systems based on AI-generated content (AIGC) applications in education.

II. LITERATURE REVIEW

A. ChatGPT

ChatGPT is a next-generation conversational natural language processing model released by OpenAI in November 2022, making AIGC a hot topic in the field of AI [15]. In addition, ChatGPT is a chatbot based on the underlying technology of generative pretrained transformer (GPT) models, which have gone through multiple iterations, evolving from GPT-1, GPT-2, and GPT-3 to the current GPT-3.5 and GPT-4. Each generation has had an exponential increase in parameters and pretraining data, and their capabilities in content generation, problem solving, and creativity are becoming increasingly similar to human cognition [16].

As a powerful AI language model, ChatGPT is now used widely in domains, such as human–computer interaction (HCI), information retrieval, and industries that require specific expertise, including medicine, finance, law, and computer code programming [17]. Previous studies have shown that ChatGPT can generate personalized responses and insightful answers for diverse purposes and domains based on the specific needs of users. The conversations are coherent, contextualized, and human-like [18]. Furthermore, ChatGPT can indirectly enhance students' critical thinking and analytical skills by detecting errors in assignments and providing immediate feedback [15].

However, researchers have gradually expressed their concerns about the continuous development of ChatGPT. For example, students may use content generated by ChatGPT as their own original work in the field of education [18], leading to academic misconduct. Students who have used ChatGPT are more prone to plagiarism compared with those who have not [19], posing serious challenges to academic integrity and fair learning for students [20]. More importantly, teachers are unable to assess students' performance when the latter use ChatGPT during the learning process, making the monitoring of student learning issues more challenging [15].

In the context of applying ChatGPT to programming education, instructors traditionally focused on hands-on coding activities to teach programming [21]. However, students often struggled to grasp programming logic and troubleshoot code errors, resulting in lower self-efficacy and motivation. ChatGPT could provide a potential solution to these challenges [22]. Some review articles suggest that ChatGPT can assist students in programming tasks, as ChatGPT can generate code based on code snippets provided by the user and optimize code performance by analyzing programming languages and algorithms. ChatGPT can also provide personalized feedback and technical suggestions to help students identify and resolve coding errors effectively [17], as well as writing more accurate and efficient code to enhance student engagement and motivation [23], [24]. Consequently, some researchers have conducted experimental studies leveraging the advantages of ChatGPT in programming

education. For instance, Yilmaz and Yilmaz [22] investigated the impact of using ChatGPT in programming education on the learning process and outcomes, and the results indicated that incorporating ChatGPT in programming education can enhance students' CT abilities, learning motivation, and programming self-efficacy. However, few studies have focused on enhancing students' CT by integrating ChatGPT into learning systems.

B. Computational Thinking

Scholars' understanding of CT varies. Wing [25] first formally proposed the concept of CT as a series of thinking activities that apply the fundamental concepts and theories of computer science to problem solving, system design, and the understanding of human behavior. Settle and Perkovic [26] summarized the seven-dimensional framework of CT as computation, communication, coordination, recollection, automation, evaluation, and design. The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association proposed an operational definition of CT and a six-dimensional framework: problem formulation, abstraction, logical thinking, algorithm application, analysis and implementation of solutions, problem generalization, and migration [27]. Computing at School proposed a five-dimensional framework for CT: algorithmic thinking, decomposition, generalization thinking, abstract thinking, and assessment skills [28]. The ISTE further refines the definition of CT to encompass a common reflection of creativity, algorithmic thinking, critical thinking, problem solving, collaborative thinking, and communication skills [29].

Many studies involve instructional design for improving CT. For example, Perez-Marín et al. [30] implemented Scratch project teaching in primary school classrooms, and the results of pre- and posttests showed that teaching computer concepts with Scratch helped to develop CT skills. The combination of educational robotics and storytelling methods provides teachers with an interdisciplinary and feasible approach to developing the CT of children [31]. However, most studies are at the K-12 level, and research involving higher education is lacking [32].

In terms of evaluating CT, various evaluation instruments have been developed, including test-based evaluation, scale-based evaluation, programming-based task/project evaluation, behavior analysis, and language expression. In the field of CT measurement, research has been conducted across various educational levels, including K-12, high school, and college. However, most studies were focused on assessing CT skills in elementary and middle schools, and doing so in high schools and colleges has received less attention, with each comprising 15% of the studies [33].

Scales of CT skills for college students are typically focused on cognitive ability and programming proficiency. For instance, Kılıç et al. [34] designed a test to assess students' existing knowledge of computer science concepts, such as variables, operations, and functions as evidence of their CT proficiency. The most widely used scale is the 29-item five-point CTS developed by Korkmaz et al. [7], which assesses students' CT skills from a cognitive perspective.

Many studies have confirmed that technology-based teaching interventions have a significant positive impact on students' CT cultivation [35]. AI-driven tools and environments can increase student engagement and motivation by interacting with them and providing them with personalized support and feedback as they learn to code. In terms of ChatGPT, researchers have found that using it in programming education can significantly improve students' CT skills [22]. However, scholars also claim that ChatGPT has limitations and potential threats; for example, overdependence on it may decrease students' ability to solve problems [22], [26], [31].

C. Scaffolding

Scaffolding has received considerable attention in education because it helps students to acquire knowledge and skills gradually in the learning process [36]. Wood et al. [37] defined scaffolding based on Vygotsky's concept of the zone of proximal development as procedural support for children or novices in problem solving or task completion that cannot be achieved alone. Scaffolding emphasizes that it will fade out when students reach a certain level of competence. Melero et al. [38] and Ma et al. [39] proposed the concept of systematic instruction scaffolding as the systematic use of technological tools or environmental designs, such as learning management systems and intelligent tutoring systems including computers or AI.

Scaffolding can be categorized into the three role dimensions of cognitive support, affective support, and transfer of responsibility [40]. Of these, responsibility transfer emphasizes the withdrawal of scaffolding support, which is also known as fading. However, it has been shown that scaffolding without fading in computer-based instruction has a greater positive impact on students [41]. Technologies such as AI and learning analytics based on big data are making huge leaps in computer-based scaffolding; this has been effective in promoting cognitive outcomes [39], [42], and some researchers have advocated using scaffolding to support the development of students' CT [8], [9].

Several researchers have conducted empirical studies to explore how CT is affected by scaffolding strategies, including instructional scaffolding, resource scaffolding support, feedback, and prompting [43], [44], [45]. Scaffolding was found to positively influence various aspects of CT [8], [9]. Yilmaz and Yilmaz [22] used ChatGPT as a scaffolding tool to explore the impact of generative AI on students' CT, and they found that ChatGPT had a positive impact on all five dimensions. Wang et al. [46] showed that scaffolding significantly affected students' problem-solving and collaboration skills in CT, which is consistent with the findings of other studies [47], [48], [49]. In addition, scaffolding improves task understanding and enhances conceptual awareness [50] by guiding students during the learning process and helping them to identify task essentials and learn about them. Scaffolding can also help students to reflect on the learning process and assess their own progress [36], enhancing their reflective thinking [51] and thus also aspects of CT. Scaffolding fades out when students can complete tasks independently once they have reached a certain level of problem solving, creativity, and other aspects of CT.

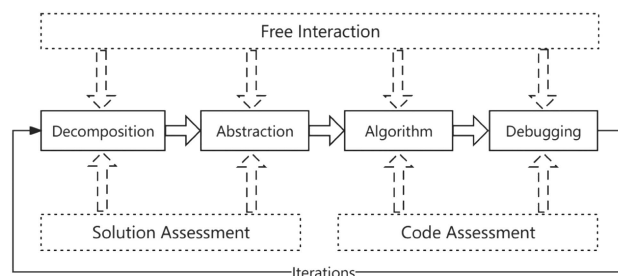


Fig. 1. Theoretical framework of the IPSSC.

Although scaffolding research has yielded many results, few previous studies were relevant to scaffolding based on generative AI for CT. Therefore, this study developed IPSSC to cultivate students' CT in programming learning to bridge the gaps in the study of intelligent scaffolding in CT. The research questions of this study are as follows.

- 1) Does IPSSC scaffold CT?
- 2) What are the students' behavioral patterns in using IPSSC?
- 3) What are the perceived benefits and challenges of IPSSC?

III. SYSTEM DESIGN

A. Theoretical Framework

Here, we propose a scaffolding framework to support the five most relevant dimensions in the CT framework, as identified by Shute et al. [14], namely, decomposition, abstraction, algorithms, debugging, and iterations. Accordingly, three scaffolding modules are designed, as shown in Fig. 1, i.e., SA, CA, and FI. In the SA module, students decompose complex tasks into smaller ones and abstract them into solution approaches for similar tasks, the focus being on cultivating students' task decomposition and abstraction abilities. The CA module is aimed at improving students' algorithm and coding abilities. Students design algorithms, write code, and continuously debug the code using this module to identify code errors and then fix them because ChatGPT can refine algorithms and code [17]. In the FI module, students can interact with ChatGPT directly to discuss the topics that have not been covered in the SA and CA modules to complete the tasks.

B. System Design

The IPSSC was developed in Visual Studio Code using a front-end and back-end separation approach. React was chosen as the front-end development framework, and Node.js was selected as the web back-end development framework. PostgreSQL was used as the database to store student accounts, interaction records, and other information. An extra GPT-3.5 backend on Amazon Web Services was deployed to receive the messages from the ChatGPT website through the official application programming interfaces (APIs) provided by OpenAI.

The interface of the IPSSC is shown in Fig. 2. The left part is the interactive area, where students' questions and ChatGPT's responses are displayed. The right part is divided into the three aforementioned scaffolding modules of SA, CA, and FI. Each



Fig. 2. Interface of the IPSSC.

module can receive students' text input and submit it to the IPSSC backend accordingly.

C. Settings of Embedded ChatGPT

The IPSSC was developed using the GPT-3.5 API to support multimodal conversations with contextual information. Table I gives an example of a teacher and students interacting in the IPSSC to complete a classroom task, where the first row contains the background of the task set by the teacher in advance. The students were given a string that contained lowercase letters and parentheses, and the algorithm had to check whether the parentheses were matched, and then remove the parentheses and return the modified string, e.g., the result of the original string “(a(b)c)d” should be “abcd.”

During the teaching process, the teacher presented the task to all the students, who had to contemplate their solutions to the task and input them into the text box in the SA module. The IPSSC facilitated communication by sending text in the SA module to ChatGPT through the official APIs provided by OpenAI and then retrieving the results. The SA module also automatically attached an additional prompt to have ChatGPT assess students' solutions, as displayed in bold text in Table I.

Students could then refine their solution or proceed with code writing by inputting the code into the text box in the CA module. After the IPSSC sent the code, along with an additional prompt as shown in bold text in Table I, to ChatGPT through APIs and retrieved the evaluation results, students could use the FI module to refine the code or request more relevant information if modifications were needed.

In brief, the design of the IPSSC provides cognitive support [40], feedback, and resource scaffolding [43] by offering students' personalized responses across three modules. These modules aim to iteratively enhance their abilities in decomposition, abstraction, algorithm design, and debugging within the CT framework [14]. Furthermore, the feedback from the IPSSC could mitigate the challenges and negative emotions associated with programming, thereby offering students affective support [40]. This study did not incorporate fading or the

TABLE I
USE SCENARIO OF INTERACTION

Role	Module	Content/Prompts
System		The task is as follows: Given a string that contains lowercase letters and parentheses, remove the parentheses and return the modified string. For example, for the string “(a(b)c)d,” the modified string should be “abcd.”
User	SA	Please assess the following solution: Create a counter variable “count” to record the number of left parentheses. Iterate through each character in the string. If it is a left parenthesis, increment “count” by 1. If it is a right parenthesis, decrement “count” by 1. After iterating through the string, if “count” is 0, it means all the left parentheses have matching right parentheses, so return true. Otherwise, return false, indicating that the parentheses are not balanced.
Assistant	SA	This solution is correct and meets the requirements of the task. However, it is important to note that this solution is only applicable to the problem of parentheses matching.
User	CA	Please assess the following code: <pre>public class RemoveParentheses { public static String remove (String s) { Stack<Character> stack = new Stack<> (); StringBuilder sb = new StringBuilder (); for (char c: s.toCharArray()) { if (c == '(') { stack.push(c); } else if (c == ')') { stack.pop(); } else { sb.append(c); } } return sb.toString(); } }</pre>
Assistant	CA	This code implements the requirements of the task, using a stack to remove parentheses. However, the code does not consider the case of mismatched parentheses.
User	FI	Please consider the case of mismatched parentheses in the code.
Assistant	FI	Sure, you can add a check for parentheses matching in the code. The specific implementation is as follows: <pre>if (! stack.isEmpty() && stack.peek() == '(') { stack.pop (); } else { // Mismatched parentheses, throw an exception throw new IllegalArgumentException("Invalid input string"); }</pre>

transfer of responsibility, as integrating an additional transfer task within the limited time of a computer laboratory session was challenging. Furthermore, existing studies have demonstrated that scaffolding, even without fading, can have a positive impact on students [41].

TABLE II
DEMOGRAPHIC INFORMATION OF PARTICIPANTS

Designation	Option	Experimental group		Control group	
		Frequency	Percent (%)	Frequency	Percent (%)
Gender	Male	7	30.4	11	47.83
	Female	16	69.6	12	52.17
Experience of using ChatGPT-like tools	Never used	5	21.7	0	0
	Used somewhat	12	52.2	7	30.43
	Regular use	6	26.1	16	69.57
Interests in programming courses	Really dislike	0	0	2	8.7
	Dislike	4	17.4	0	0
	Neutral	9	39.1	8	34.78
	Like	7	30.4	13	56.52
	Really like	3	13.0	0	0
Preferred media type for learning	Text	3	13.0	3	13.04
	Video or photo	17	74.0	15	65.22
	No particular preference	3	13.0	5	21.74

IV. METHODOLOGY

A. Research Settings and Participants

This study involved a course on Data Structures and Algorithms at a university in Southwest China. Twenty-six sophomore undergraduates in Educational Technology participated in the experimental group during the spring semester of 2023, and 23 of them completed the demographic information survey, as presented in Table II. In addition, another 26 sophomore undergraduates in Educational Technology (14 male and 12 female) participated in the control group during the fall semester of 2023. Similarly, 23 students completed the demographic information survey listed in Table II.

B. Instruments

This study used the CTS as the pre- and posttest instrument. A learning experience questionnaire was designed to investigate the perceived experiences of students about the IPSSC with nonscaled questions. Finally, a semistructured interview outline was developed to gain insights into the effectiveness of students' use of IPSSC in the three scaffolding modules of SA, CA, and FI, as well as its impact on the overall learning outcomes of the students. The instruments used in the study are explained in more detail as follows.

1) *Computational Thinking Scale*: This study used the CTS developed by Korkmaz et al. [7] to assess the changes in students' CT ability via pre- and posttests. This scale was chosen for its focus on measuring the cognitive CT ability of college students, aligning with the purpose of this study. The scale has 29 items for the five subfactors of creativity, algorithmic thinking, cooperativity, critical thinking, and problem solving. The scale has five Likert-type evaluation levels, and the higher the score, the higher the thinking ability of the student. The Cronbach's alpha reliability coefficient for the whole scale is 0.822, and

for the five factors, it is 0.843, 0.869, 0.865, 0.784, and 0.727, respectively.

2) *Learning Experience Questionnaire*: A questionnaire was designed to investigate students' experiences and the perceived effectiveness of using IPSSC. The questionnaire had three parts, i.e., basic information, learning effect, and personal attitude, with a total of 17 items. Cronbach's α coefficient was used to measure the reliability of the data. The Cronbach's α coefficient for the learning effect and personal attitude as a whole was 0.774, indicating acceptable reliability. The Kaiser-Meyer-Olkin (KMO) value was 0.660 (>0.6), and the Bartlett test showed a p -value of 0.000, indicating that the validity of the questionnaire was within an acceptable range. Basic information was aimed at obtaining students' prior knowledge and learning styles, learning effect was aimed at investigating the perceived effect of IPSSC, and personal attitude was aimed at investigating learners' views and preferences about the application of IPSSC.

3) *Interview Outline*: This study used a semistructured interview outline to investigate further the specific process of students using the proposed system. A semistructured interview allows the interviewer to engage in deep communication with the interviewee when necessary; the interviewer can extend the questions flexibly based on the interviewee's responses and dig into the interviewee's thoughts more deeply when the interviewer finds worthy viewpoints. The outline had six questions covering four parts: the general perception of IPSSC, the function of the three modules of IPSSC, the perceived effectiveness, and the advantages and disadvantages of using IPSSC.

C. Procedure and Data Collection

The Data Structures and Algorithms course comprised 16 theoretical sessions and seven computer laboratory sessions throughout the semester. Students in both the experimental and control groups were asked to complete the CTS pretest before the first computer laboratory session. Each student was required to enter a unique secret number known only to them, to ensure anonymity in the pretest while allowing researchers to track each student's responses in the posttest.

In the experimental group, the IPSSC was used to scaffold the students' programming during all computer laboratory sessions. Each student was required to finish a coding task of the course independently in the first four sessions and collaboratively with another student in the last three sessions. Students could write a solution to solve the problem in each task and then paste it into the SA module of the IPSSC to get feedback from ChatGPT. The students could then implement the algorithm and send the code to the CA module to obtain specific feedback on the code. The students could also use the FI module whenever they wanted.

In the control group, students were asked to complete the same tasks independently in the first four sessions and collaboratively in the last three sessions. Instead of using IPSSC to support them in completing the tasks, students in the control group were only able to ask the teacher for help if difficulties arose.

After all the computer laboratory sessions had ended, students in the control group were asked to complete only the CTS posttest. In contrast, students in the experimental group were

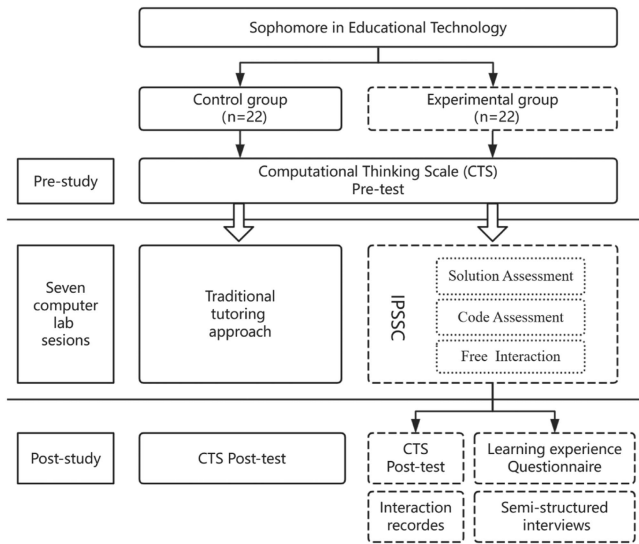


Fig. 3. Study procedure.

asked to complete both the CTS posttest and the learning experience questionnaire, using the secret numbering system again for matching. Also, nine students in the experimental group were selected for interview based on the number of their interaction records in the IPSSC. The study procedure is shown in Fig. 3.

D. Data Analysis

This study employed a pre/posttest design to compare students' CTS scores before and after using IPSSC. First, the Shapiro–Wilk test was utilized to assess whether the subjects conformed to a normal distribution. Subsequently, the authors examined whether the variances of the two independent samples were equal. The independent t -test (IT) or Mann–Whitney U test (M-W) was conducted to determine significant differences between the pretest or posttest scores of the experimental group and the control group at the levels of the entire scale and its five dimensions, depending on whether the two groups' data adhered to both normal distribution and homogeneity of variance. Following this, a paired-samples t -test (PT) or Wilcoxon test (WT) was applied to ascertain significant differences between pre- and posttest scores at the levels of the entire scale and its five dimensions for both the experimental and control groups, depending on whether the two sets of data met the criteria of normal distribution and homogeneity of variance. Moreover, descriptive statistical analysis was used to show the results of basic information, learning effects, and personal attitudes of the participants from the learning experience questionnaire for the experimental group.

Second, descriptive statistics, lag sequential analysis, and K -means clustering were used to analyze the causal relationships and interactive patterns in the interaction between students and ChatGPT in the IPSSC. Descriptive statistics were used to count the number of interaction records. Lag sequential analysis is typically used to visualize and analyze behavioral data in HCI, aiming to examine whether there are statistically significant

sequential relationships between various behaviors [52], [53]. In this study, the lag sequential method was used to analyze the sequences of students' interactions in the system. K -means clustering analysis was used to mine the interaction patterns.

Finally, this study adopted the semistructured interview method, which refers to informal interviews conducted according to a rough outline of the interview [54]. Semistructured interviews can further obtain students' authentic thoughts and insights, thus enriching and improving the credibility and accuracy of the findings. Following the principles of stratified and purposive sampling, this study interviewed nine students from three groups: students interacting with ChatGPT in the IPSSC more often (more than 300 records), moderately (140–150 records), and less often (30 records or fewer). The different levels of the students represented different views about communicating with ChatGPT in the IPSSC and helped the researcher to obtain richer insights.

Nvivo, a qualitative analysis research tool, was used to analyze the interview transcripts by coding the sentences, including important ideas in the transcripts. Codes with similar meanings were combined, and codes containing multiple meanings were split into multiple codes. All codes were then categorized based on the interview questions, and the number of excerpts attached to each code was counted to demonstrate the popularity of each idea. After one author of the study extracted all codes from the transcripts, another author verified the coding by reviewing all codes and the attached excerpts and then discussed with the previous author to reach an agreement on each code in the coding scheme.

V. RESULTS AND FINDINGS

A. Does IPSSC Scaffold CT?

All students in both the experimental group ($n = 26$) and the control group ($n = 26$) were required to enter a unique secret number known only to the student during both the pretest and posttest phases to ensure anonymity in the surveys. Twenty-two pairs of secret numbers were matched between the pretest and posttest in both the groups. Therefore, 22 participants for each group were selected for the data analysis of CTS.

The Shapiro–Wilk test was applied to determine that the sample followed a normal distribution when $p > 0.05$; otherwise, it deviated from normality. The results of the normality test indicated nonnormal distribution in three dimensions for the control group, including CT ability ($p = 0.002$), critical thinking ability ($p = 0.008$), and problem-solving ability ($p = 0.007$). Furthermore, the data in the dimension of cooperation ability did not adhere to a normal distribution for both the experimental group ($p = 0.009$) and the control group ($p = 0.023$). However, the variances were consistent across all samples. Nonparametric methods were applied to examine differences between groups and within groups for the data that did not conform to normal distribution.

Tables III and IV present a comparative analysis of intragroup and intergroup differences in the CTS dimensions. The results of the IT and M-W on pretests show that there were no significant differences between the experimental group and the control

TABLE III
COMPARATIVE ANALYSIS OF INTRAGROUP DIFFERENCES ON THE CTS
DIMENSIONS

	Group	<i>n</i>	Pre / post tests	Method	Mean, SD, <i>t</i> / M (P25, P75), Z			<i>p</i>
Computational thinking	EG	22	Pretest	PT	99.64	6.89	–	0.015*
			Posttest		102.68	6.88	2.66	
	CG	22	Pretest	WT	101.00	107.75	–	0.465
			Posttest		103.50	108.50	0.73 ^b	
Creativity	EG	22	Pretest	PT	29.27	2.95	–	0.006**
			Posttest		31.18	2.89	3.08	
	CG	22	Pretest		30.86	2.75	–	0.688
			Posttest		31.18	3.28	0.41	
Algorithmic thinking	EG	22	Pretest	PT	18.86	4.06	–	0.006**
			Posttest		20.59	4.50	3.07	
	CG	22	Pretest		22.55	4.37	–	0.559
			Posttest		21.23	2.49	0.59	
Cooperativity	EG	22	Pretest	WT	12.75	17.00	–	0.565
			Posttest		13.00	18.25	0.58 ^b	
	CG	22	Pretest		12.75	16.00	–	0.090
			Posttest		14.75	17.00	1.69 ^b	
Critical thinking	EG	22	Pretest	PT	17.45	2.54	–	0.316
			Posttest		17.86	2.87	1.03	
	CG	22	Pretest	WT	17.00	20.00	–	0.203
			Posttest		18.00	22.00	1.27 ^b	
Problem solving	EG	22	Pretest	PT	18.82	3.69	–	0.015*
			Posttest		17.27	3.30	2.66	
	CG	22	Pretest	WT	16.75	22.00	–	0.536
			Posttest		16.75	21.25	0.62 ^c	

Note: * $p < 0.05$, ** $p < 0.01$. ^bBased on negative ranks. ^cBased on positive ranks.

group at all dimensions, including CT ($p = 0.057$), creativity ($p = 0.084$), algorithmic thinking ($p = 0.193$), cooperativity ($p = 0.317$), critical thinking ($p = 0.083$), and problem solving ($p = 0.548$).

Regarding the general CT dimension, the results from the posttest IT and the M-W (see Table IV) revealed significant differences in CT abilities between the experimental and control groups ($p = 0.032$). In addition, students in the experimental group showed significant improvement in CT abilities from pretest to posttest ($p = 0.015$), while no significant differences were observed between the pretest and posttest results in the CT abilities of students in the control group (refer to Table III). These findings suggest that employing the IPSSC as a scaffold in the

TABLE IV
COMPARATIVE ANALYSIS OF INTERGROUP DIFFERENCES ON THE CTS
DIMENSIONS

	Pre / post tests	Group	<i>n</i>	Method	Mean, SD, <i>t</i> / M (P25, P75), Z			<i>p</i>
Computational thinking	Pretest	EG	22	IT	99.64	6.89	–1.95	0.057
		CG	22		104.18	8.42		
	Posttest	EG	22	M-W	102.00	107.00	–2.14	0.032*
		CG	22		103.50	108.50		
Creativity	Pretest	EG	22	IT	29.27	3.19	–1.77	0.084
		CG	22		30.86	2.75		
	Posttest	EG	22	IT	31.18	2.89	0.00	1.000
		CG	22		31.18	3.28		
Algorithmic thinking	Pretest	EG	22	IT	18.86	4.06	–1.32	0.193
		CG	22		20.55	4.37		
	Posttest	EG	22	IT	20.59	4.50	0.06	0.565
		CG	22		21.23	2.49		
Cooperativity	Pretest	EG	22	M-W	12.75	17.00	–1.00	0.317
		CG	22		12.75	16.00		
	Posttest	EG	22	M-W	13.00	18.25	–0.87	0.385
		CG	22		14.75	17.00		
Critical thinking	Pretest	EG	22	IT	17.45	2.54	–1.77	0.083
		CG	22		18.68	2.01		
	Posttest	EG	22	M-W	16.00	20.00	–2.06	0.039*
		CG	22		18.00	22.00		
Problem solving	Pretest	EG	22	IT	18.82	3.69	–0.60	0.548
		CG	22		19.45	3.28		
	Posttest	EG	22	M-W	15.00	20.00	–1.35	0.174
		CG	22		16.75	21.25		

Note: * $p < 0.05$, ** $p < 0.01$.

Data Structures and Algorithms course significantly enhances students' CT skills overall.

The results for the dimensions of creativity ($p = 0.006$), algorithmic thinking ($p = 0.006$), and critical thinking ($p = 0.039$) show significant improvements in either intragroup or intergroup differences. Surprisingly, the students' problem-solving skills in the experimental group decreased significantly from pretest to posttest ($p = 0.015$), although there was no significant difference in the results of the intergroup posttest comparison, which may be attributed to the small sample size.

B. Patterns of Interaction With ChatGPT

Fig. 4 displays a weekly graph of the HCIs of the experimental group across seven sessions of the study. The graph illustrates a rapid increase in HCIs from 613 to 884 between the first and second weeks. This surge may be attributed to the upgrade of the IPSSC from GPT-3 to GPT-3.5 in the second week, enabling learners to engage in multiturn conversations with ChatGPT. There was no computer laboratory session on May 1 due to the national labor holiday.

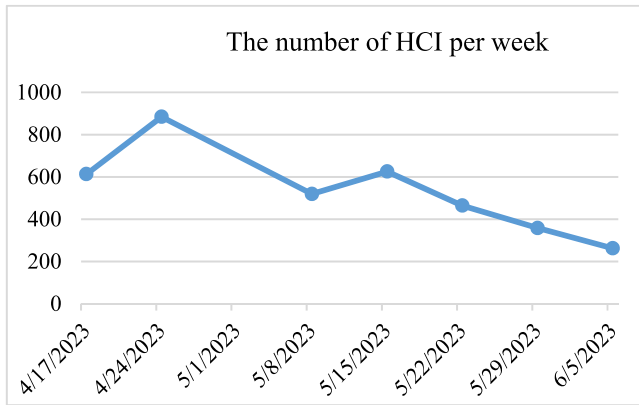


Fig. 4. Number of HCIs per week.

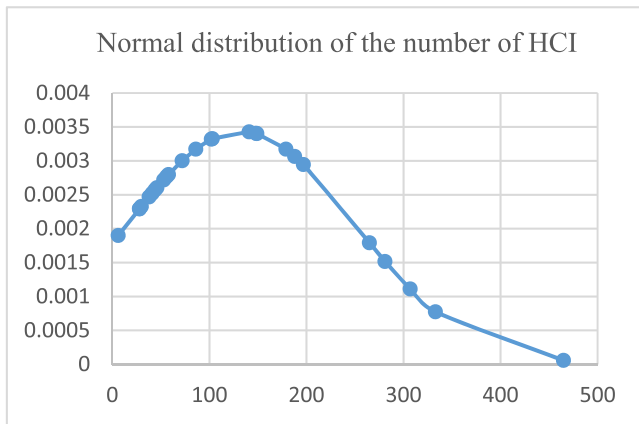


Fig. 5. Normal distribution plot of the number of HCIs.

TABLE V
BEHAVIOR FREQUENCY OF INTERACTION RECORDS

Code	Module Name	Relevant Records	All Records
SA	Solution Assessment	352	506
CA	Code Assessment	161	174
FI	Free Interaction	907	1230
Total		1420	1910

However, the overall number of records in the remaining weeks was lower than in the first and second weeks. This could be attributed to the diminishing novelty as learners became more familiar with ChatGPT in the IPSSC. Alternatively, students might have become more proficient in using the IPSSC to obtain the desired results with fewer interactions.

A normal distribution diagram (see Fig. 5) was created to show the central tendency of the total number of HCIs and the dispersion of individual student interactions. The diagram indicates that the highest probability of interaction counts falls within the range of 100–200 per student, and the number of HCI records is approximately normally distributed.

The interaction records for each module were counted, as given in Table V. Records that were relevant to the course

TABLE VI
BEHAVIOR FREQUENCY IN FREE INTERACTION

	Encoding	Behavior types	Example	Number
Students Behaviors	DC	Decomposition	<ol style="list-style-type: none"> 1. Define a customer class that includes two attributes: arrival time and required time for service. 2. Define a queue to simulate the waiting area. Whenever a new customer arrives, add them to the end of the queue. 3. Define an array or list to represent the status of three counters, with the initial state being all vacant. 4. Simulate the entire process by simulating time, checking every minute. 5. Calculate the average waiting time for all customers, which is the sum of all customer waiting times divided by the total number of customers. 	128
	AB	Abstraction	Write a BankSimulation class in Java that implements a functionality to allow customers to queue in the waiting area and assign them to a counter for service when there is a vacant counter. Calculate the average waiting time for all customers during the simulated time period. Use an array or list to represent the status of three counters (vacant or busy).	219
	AL	Algorithms	<pre>public BankSimulation (int simulationTime, int arrivalTimeRange, int serviceTimeRange) { this.simulationTime = simulationTime; this.arrivalTimeRange = arrivalTimeRange; this.serviceTimeRange = serviceTimeRange; this.counters = new int[3]; this.waitingQueue = new LinkedList<>(); this.totalWaitTime = 0; } public void simulate() { double averageWaitTime = (double) totalWaitTime / (double) waitingQueue.size(); System.out.println("averagewaittime:" + averageWaitTime + "minute"); }</pre>	192
	DB	Debugging	Why is the result of running the program an average waiting time of 0.0 minutes?	329
	IC	Irrelevant Categories	Content unrelated to this course.	39

content were then identified and counted, and the results show that most records were relevant to the course content (1420/1910). Students used the FI module the most (907/1230) and the CA module the least (161/174), but there were only a few irrelevant records in the CA module. According to the results, students used the FI module the most in the HCI process, although they made use of the other two to some extent.

The interaction records for each component of CT in the FI module were counted, as given in Table VI. The results show that the interaction topics covered all components of CT, especially debugging (329), abstraction (219), algorithms (192), and decomposition (128).

The sequential interaction patterns were further analyzed using lag sequential analysis. The interaction sequences of each student were imported into GSEQ5, a lag sequential analysis software package, and Table VII gives the transitions from the previous behavior to the next. For example, the datum in the first

TABLE VII
BEHAVIOR FREQUENCY CONVERSION TABLE

	SA	CA	FI	Total
SA	225	33	91	349
CA	25	61	71	157
FI	79	67	744	890
Totals	329	161	906	1396

TABLE VIII
ADJUSTED RESIDUAL TABLE

	SA	CA	FI
SA	*20.79	-1.40	-17.55
CA	-2.40	*11.38	-5.48
FI	-17.15	-6.21	*19.41

* Z-score >1.96, indicating statistical significance.

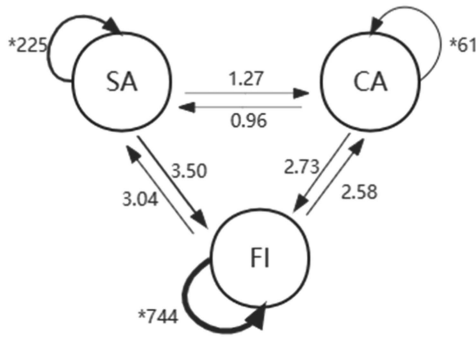


Fig. 6. State transformation diagram.

row and third column of Table VII represents students engaging first in SA and then proceeding to FI, with a behavior sequence represented as SA-FI and a transition count of 91.

An adjusted residual table (see Table VIII) was generated to facilitate the calculation of residual values for the frequency of each sequence, and the values in this table represent the residual values (Z-scores). According to the theory of lag sequential analysis, the sequence relationships between each behavior have statistical significance when the Z-score is greater than 1.96. Table VII shows that three behavior sequences reached significance, namely, SA-SA, CA-CA, and FI-FI.

A state transform diagram (see Fig. 6) was created to depict the transitions of students' behaviors throughout the learning process. The learning behaviors SA, CA, and FI all have arrows pointing to themselves, and their Z-scores are all greater than 1.96, indicating statistical significance. This suggests that students tend to repeat certain actions over and over again in the learning process, which could be evidence of deep learning by exploring better answers via multiturn interaction.

The Z-score for the SA-SA sequence is the highest, indicating that students are more inclined to follow steps in their learning activities initially. However, the Z-score for the CA-CA sequence is much lower compared to the other two behavior sequences. This suggests that learners may encounter difficulties in code writing during CA and consequently do not persist in communicating with ChatGPT in CA. As a result, they exhibit

TABLE IX
K-MEANS CLUSTERING ANALYSIS

	Interaction-oriented	Solution-oriented	Solution-code-oriented	Solution-interaction-oriented
SA	12.13	89.19	44.54	32.18
CA	8.16	10.74	24.22	6.21
FI	87.62	3.77	6.29	49.63
Number of participants	12	3	4	7

a preference for interacting with ChatGPT via the FI module to seek answers to their questions.

Analyzing the values on the arrows among the three sequences shows that the average number of behavior transitions for learners is highest for the sequences SA-FI and FI-SA, and the values are relatively consistent. This indicates that students tend to complete SA before engaging in FI when interacting with the AI. The sequences SA-CA and CA-SA have the lowest average number of behavior transitions, suggesting that students are less inclined to follow a sequential order for CA after completing SA. Therefore, it can be concluded that students tend to exhibit behavior sequences such as SA-FI, FI-SA, CA-FI, and FI-CA in their learning process.

Furthermore, this study used K-means clustering analysis to classify the students' personal preference types into four categories: interaction-oriented, solution-oriented, solution-code-oriented, and solution-interaction-oriented (see Table IX). The K-means clustering analysis used in this study could determine the classification of samples based on a small number of known clusters. The authors attempted to classify the subjects into three, four, and five categories, and the results indicated that four categories are the most appropriate for showing the distinctions among the categories, and no category could be further subdivided. The naming of the four proposed categories was based on which one or two modules were used the most.

As given in Table IX, 12 students in the interaction-oriented category preferred to use the FI module to solve problems, accounting for 87.62% of the total proportion among the three modules. Three students in the solution-oriented category were more inclined to use the SA module to solve problems, accounting for 89.19% of the total proportion. Four students in the solution-code-oriented category mainly used the SA and CA modules. The reason could be that some students prefer to learn step by step in the learning process, constantly improving their solutions in the SA module, and then completing the code writing. Because of the high level of participation in the first two modules, students can solve the difficulties they encounter on their own, so there are fewer utterances in the FI module. Finally, seven students in the solution-interaction-oriented category were more inclined to solve problems in the FI and SA modules. Combining the analysis of the state transform diagram, these students were more inclined to use the FI module after the SA module to solve problems.

TABLE X
ANALYSIS OF LEARNING EXPERIENCE QUESTIONNAIRE

Items of the Survey	Options					
	Strongly disagree	Disagree	Neutral	Agree	Strongly agree	
Perceived Effectiveness	I found the information provided by IPSSC helpful when solving the problems I faced in the course.	0%	0%	4.4%	82.6%	13.0%
	I found the information provided by IPSSC to be detailed and understandable.	0%	0%	34.8%	65.2%	0%
	In program assignment, I think IPSSC can give a reasonable evaluation.	0%	4.35%	52.2%	39.1%	4.6%
	In the code assignment, I found IPSSC helpful in correcting and commenting on the code.	0%	4.35%	52.2%	34.8%	8.7%
	In a free interaction, I think IPSSC can provide solutions or ideas to solve the problem.	0%	0%	30.4%	56.5%	13.0%
	In group collaboration, you see IPSSC playing a role.	0%	0%	26.1%	56.5%	17.4%
	Personal attitude	My acceptance of incorporating IPSSC's Data Structures course learning.	0%	0%	26.1%	47.8%
I would like to continue to use IPSSC to assist me in the future.		0%	4.35%	13.1%	47.8%	34.8%

TABLE XI
CODING SCHEME FOR INTERVIEW DATA

Code schemas	Categories	Numbers	Original codes	Number of references	
Overall experience	Positive experience	6	ChatGPT stores a large amount of knowledge	1	
			Code can be generated instantly upon request	5	
	Negative feedback	18	Need skills to ask the right questions	3	
			Overdependent on the system	6	
			The system is instable	3	
			Inability to answer complex questions	4	
			The answers are not innovative	2	
Effectiveness	Positive feedback	25	Inspire learning motivation	3	
			Deepen understanding of knowledge	4	
			Stimulate thinking	4	
			Improvement of programming skills	6	
	Negative feedback	0	-	Improving the efficiency of access to information	8
					0
					0
Solution Assessment	Positive feedback	6	The ability to optimize ideas	6	
	Negative feedback	5	The suggested modifications are not accurate	2	
			The given solutions are too complex than the class requires	3	
Code Assessment	Positive feedback	22	Relatively accurate assessment of codes	6	
			Problems in the code can be identified	5	
			The ability to optimize code in details	11	
	Negative feedback	6		The given answers require further organized	3
				Repeat answers	3
				Helpful for understanding code	1
				Helpful for understanding concepts	5
Free Interaction	Positive feedback	19	Helpful for writing code	13	
				5	
	Negative feedback	5		Inability to understand students' intention fully	5
				Tool to support learning	3
Role of the module	11		Teacher	4	
			Databases	4	

C. Perceived Benefits and Challenges

This study used both a questionnaire and interviews to investigate the perceived benefits and challenges of the IPSSC. First, the researcher conducted descriptive statistical analysis on the questionnaire survey results (see Table X). Next, the researcher used the Nvivo12 software to conduct qualitative analysis of the interview results. The interview content was divided into five parts, namely, overall feelings, perceived effect, SA module, CA module, and FI module. Positive and negative effects were compiled under each of the five parts, and the relevant content was coded under this categorization, with coding specifics, as given in Table XI. The results of the analysis are as follows.

1) *Perceived Learning Effect*: The results of the questionnaire showed that more than 60% of the students could understand the information provided by IPSSC well, and more than 90% reported that the information provided by the IPSSC was beneficial for solving problems, indicating that the IPSSC can solve most problems for students. The coding of the interview content showed that the interviewed students believed that the IPSSC could improve their learning in Data Structures and Algorithms to different degrees, including enlightening thinking, improving programming ability, and improving information acquisition efficiency.

2) *Solution Assessment*: The questionnaire results showed that 43.5% of the students believed that the IPSSC could give a

reasonable evaluation of the solutions written by students. Moreover, students could gain thinking inspiration from the feedback generated by ChatGPT. However, the feedback provided by ChatGPT in the IPSSC had some drawbacks. For example, the optimized solution given by the IPSSC was sometimes more complex than the student expected for code implementation and did not meet the current teaching progress and difficulty. Consequently, this might increase the cognitive load of students and cause unnecessary difficulties in problem solving.

3) *Code Assessment*: The questionnaire results showed that 43.5% of the students thought that the IPSSC was helpful for identifying and correcting the errors in the code by using the CA module, while 52.2% remained neutral. The content analysis of the learners' interview responses revealed that the number of positive feedback instances from the students regarding the CA module was much higher than that of negative feedback. The interviewed students reported that in the CA module, the IPSSC could identify the problems in the code and suggest how

to modify and optimize it. However, students sometimes needed to rework the answers from the IPSSC to make the code run successfully.

4) *Free Interaction*: The questionnaire results showed that 69.6% of the students believed that the FI module in the IPSSC could provide useful feedback on solutions or ideas to finish the tasks in the computer laboratory sessions. The coding analysis of the interview transcripts also showed that the function of FI is powerful. Students generally believed that ChatGPT in the IPSSC played three roles in their learning: teacher, database, and learning aid. Students reported that the IPSSC could help them understand the code, understand concepts, and even help them write code when using the FI module. However, some students reported that ChatGPT in the IPSSC sometimes could not fully understand the students' difficulties if the questions were not asked appropriately. Therefore, whether the students could get the desired answer mostly depended on their questioning strategies.

5) *Attitude and Feelings*: The results of the questionnaire showed that 73.9% of the students agreed with the integration of IPSSC into the Data Structures and Algorithms course, while 26.1% were neutral. Moreover, 82.6% of the students were willing to continue to use the IPSSC to assist them in their learning. Some students believed that the IPSSC was a large knowledge base, which could improve the efficiency of information acquisition. The IPSSC could improve programming ability and inspire thinking to a certain extent through reasonable teaching design. Students also reported some drawbacks of the tools. For example, most students worried that they might over-rely on the IPSSC and, thus, fail to finish similar tasks without its support. Also, the answers from the IPSSC had low innovation because ChatGPT only synthesized the existing learning materials on the Internet. In addition, ChatGPT still could not answer complex questions accurately to a certain degree.

VI. DISCUSSION

A. *Affordance of ChatGPT on Scaffolding CT*

The main purpose of this study was to investigate the affordance of ChatGPT on scaffolding CT via the CTS analysis, the interaction patterns, the questionnaire, and the interview transcripts. The CTS findings are generally aligned with those in [8], in which incorporating ChatGPT in programming education was shown to enhance students' CT abilities, especially in the dimensions of creativity, algorithmic thinking, and critical thinking.

Creativity involves introducing new relations and forming new combinations from one or more concepts in the mind for the purpose of looking at events from different aspects [55]. The paired *t*-test results showed that the creativity of students in the experimental group improved significantly, and this result is supported by the students' interview reports, such as those reporting that the IPSSC inspires learning motivation, deepens the understanding of the relevant knowledge, and optimizes students' ideas, as given in Table XI. Therefore, the IPSSC can inspire students' thinking and enhance their creativity by

optimizing their problem-solving ideas in the SA module and helping them understand the concepts involved in the course in the FI module.

Algorithmic thinking is the ability to think about, understand, apply, evaluate, and create algorithms [56]. The *t*-test results indicate a significant improvement in the algorithmic thinking of students in the experimental group. A possible reason for this improvement is that the IPSSC can assist students in identifying problems in their code and optimizing it to enhance their understanding of algorithms. The interview results revealed that the CA module could assess code somewhat accurately, identify problems within it, and optimize it in detail, thereby reinforcing the aforementioned statements. In addition, the FI module can aid in understanding both code and concepts, enabling students to write code, as illustrated in Table XI.

Critical thinking is viewed as the skill of determining assumptions, hidden beliefs, values, and attitudes [7]. The M-W revealed that posttest scores for critical thinking skills in the experimental group were significantly higher than those in the control group, while no significant differences were found in the pretests between the two groups. This may be due to ChatGPT not providing entirely accurate responses, as indicated by students' interview reports in Table XI. Students still need to critically discern whether the provided responses are correct to complete the tasks.

To date, few studies have investigated the interaction patterns between students and ChatGPT. The present findings regarding interaction patterns show that most students kept using the proposed system to support them to finish the tasks via multiple turns of conversation with ChatGPT, although different students tended to use the scaffolding modules in the proposed system in different ways. Also, the conversations in the FI module basically covered all the CT components in [8], showing that the proposed system can help students to improve their CT in most aspects, including decomposition, abstraction, algorithms, and debugging.

Other findings from the questionnaire and interviews show that ChatGPT in the proposed system can provide instant feedback and evaluate the solution and code in the tasks accurately to some degree. The students also reported that the IPSSC increased their motivation, improved their understanding of the algorithms and code, helped them to search for relevant information, and inspired them to think more openly and deeply. As a result, most of the students had positive attitudes toward the tools and were willing to use them in the future.

These findings show that the IPSSC is more than a substitute for teacher evaluations. In addition to providing assessments anywhere and anytime, the IPSSC can also provide personalized guidance for individual student to solve encountered problems and optimize task solutions, algorithms, and code via multiple rounds of interaction.

However, the findings also show some drawbacks of using ChatGPT in programming learning. For example, the problem-solving skills of students in the experimental group decreased significantly, which is in conflict with the findings in [8]. When the authors conducted in-depth interviews with students on their experiences of using the IPSSC, most students mentioned that

they directly turned to the IPSSC for help when they had difficulty in a programming task, because the IPSSC could generate code feedback based on the requests in most cases. Most students reported that they easily became dependent on it, leading to a lack of their own thinking and simply running the code provided by the IPSSC. Meanwhile, Table IX shows that most students fell into the pattern of being interactive-oriented, i.e., they preferred to use the FI module directly. This finding supports the interview conclusion that students often turned to the IPSSC for help and occasionally overlooked the steps to refine their own solution and code.

Studies have shown that learning programming enhances students' problem-solving skills [57]. However, this process requires learners to actively engage their minds in understanding the fundamental concepts of programming, as well as in analyzing, organizing, implementing, and evaluating code [58]. In this study, according to the quantitative and qualitative results from interaction records and interview data, students primarily used ChatGPT in the FI module to directly obtain code, subsequently failing to engage in active thinking or to independently apply fundamental programming concepts to their solutions. Consequently, students' problem-solving skills declined due to an overreliance on ChatGPT, rather than improving.

Employing ChatGPT simplifies the process of obtaining answers or information, which can adversely affect students' capacity to independently solve problems [2]. In the forthcoming era of human-machine symbiosis, education will increasingly depend on AI and must prioritize the development of higher level learning outcomes [59]. Therefore, the design of learning content and tasks continues to be a primary concern [11], [60]. In brief, we assert that integrating intelligent tools, such as ChatGPT, into educational practices does not ensure a positive effect on all aspects of learning. Some skills, such as problem solving in CT, may be negatively impacted without further design, so the instructional design is the key to determining the impact of the AI tools on instruction or learning.

B. Demands on Further Design of Scaffolding

Here, four scaffolding strategies are proposed to refine the design of IPSSC with the aim of further improving CT in programming learning, especially regarding problem solving.

1) *Scaffolding With Fading*: Fading is considered as an important component in the framework of scaffolding [40], but whether scaffolding should be implemented with or without fading in computer-based instruction is still controversial [36]. This study used a scaffolding design without explicit fading because of the time limitation of the computer laboratory sessions. Nonetheless, Fig. 5 shows that the number of weekly HCIs decreased gradually. Although a possible reason for this is novelty effect (i.e., students gradually become less interested in a new technology as they become more familiar with the tools), a more reasonable explanation is that the students became more competent coders and, therefore, did not need ChatGPT as much, i.e., the scaffolding of IPSSC faded implicitly.

As previously mentioned, the students' problem-solving skills, as measured by the CTS, decreased significantly. One possible reason for this decline could be that they had become accustomed to relying on ChatGPT in the IPSSC to solve problems. As a result, they might have lacked confidence when confronted with similar tasks without the assistance of AI tools. Therefore, it would be intriguing to investigate whether an explicit fading design plays an important role to better support students in solving problems by themselves. For example, the teacher could give students a relatively small task to solve with the support of ChatGPT embedded in the AI tools and then ask them to finish a similar task without the support of ChatGPT. The students would be expected to generalize and transfer their knowledge to solve similar tasks. The fading design would also be aligned with the generalization component in the CT framework proposed by Shute et al. [14], which has not yet been integrated into the design of this study.

2) *Personalized Scaffolding*: Analysis of the interaction patterns between students and ChatGPT in the IPSSC shows that different students preferred different patterns, including interaction-oriented, solution-oriented, solution-code-oriented, and solution-interaction-oriented. These findings align with the rationales of personalized learning, where each individual student has different learning demands and learning styles [61]. Therefore, although ChatGPT3.5 can track conversation history and provide contextual answers, it is important to examine further the learning style and zone of proximal development of each student and provide support accordingly.

3) *Scaffolding to Ask the Right Questions*: According to the students' reports, it is important to ask ChatGPT the right questions to get more accurate and detailed answers. However, most students have limited knowledge for framing the questions well on their own. A possible solution is to provide students with more guidance on the strategies of querying ChatGPT in an appropriate way. Thus, two types of scaffolding could be designed: 1) process scaffolding, which scaffolds the students to follow the steps in the learning process toward CT [62] and 2) prompt scaffolding, which support students to phrase better prompts as questions [63].

4) *Human-Computer Collaborative Scaffolding*: Many students reported that ChatGPT is still not smart enough to understand their needs, and a better solution could be to incorporate the instructor into the scaffolding process [64]. Although the IPSSC, as currently designed, can record students' conversations with ChatGPT for the teacher to review, how the teacher can scaffold the learning process collaboratively with AI tools embedding ChatGPT should be examined further. For example, while ChatGPT has the advantages of responding instantly and covering every student, its ability to understand complex questions is still limited. Teachers can better understand students' needs with these tools. However, guiding individual students takes time, and it is difficult to cover every student individually in a session, especially when the class has many students. Therefore, the tools can leverage the advantages of both humans and machines by providing functions to identify the students who have the most difficulties in conversing with ChatGPT, and teachers can then intervene accordingly. Moreover, it is necessary to investigate

further how the teacher and the AI tools could better scaffold multiple students to finish a task because the CTS results show that the increase in cooperativity was not significant.

C. Demands on the Next Generation of AIGC

The present findings from the questionnaire and interviews also provide insights into designing AIGC models for better supporting CT and programming learning.

First, most students reported that ChatGPT3.5 was still not smart enough to understand all of their demands, especially when the questions were complex. The responses of ChatGPT were then inaccurate to some degree in this case, and students felt that ChatGPT only synthesized the answers from the Internet. In addition, the answers of ChatGPT were repeated sometimes even when the students asked similar but different questions. It is then necessary to keep improving the intelligence of AIGC models to improve the answer quality for instructional purposes.

Another demand on the AI tools is to integrate emotion recognition to provide responses accordingly, because some students mentioned that ChatGPT could not understand their emotions during the learning process. For example, AI tools could offer more detailed information if the system visually detects that a student is confused by the previously generated text. Otherwise, the AI tools only need to provide concise answers thereafter.

Finally, it is important to have the AI tools provide responses in rich media to help students better understand the concepts, flows, or algorithms in programming learning. As shown in the students' demographic information in Table II, 74.0% of those in the experimental group and 65.22% in the control group preferred video and picture learning materials, while only about 13% in both groups preferred text learning materials. However, ChatGPT3.5, as used in this study, could only interact with learners in text form, which may have influenced the learning outcomes accordingly.

VII. CONCLUSION

This article proposed an IPSSC following the theoretical framework of CT and scaffolding. Three modules were designed in the proposed system, i.e., SA, CA, and FI. The findings of the mixed-method study showed that the IPSSC was effective for improving the students' CT generally, and most had positive attitudes about the proposed system. However, the results also showed that the problem-solving skills of the students in the experimental group decreased significantly after the study. Four further scaffolding strategies were then proposed to guide the design of the tools embedding ChatGPT to foster students' CT and programming learning.

Although this study has provided initial findings to enrich the theories and practice of AI-supported programming learning, it has limitations due to the short preparation period. For instance, the scaffolding strategy used in this study is still in an exploratory stage, and the design of the IPSSC could be further refined based on the four proposed scaffolding strategies. The sample size of the study is also limited.

The future plan resulting from this study is to refine the IPSSC following the proposed scaffolding strategies. A design-based

research will then be conducted to examine the new design and relevant design principles on a larger scale of participants.

REFERENCES

- [1] E. K. Jelovac and D. M. McLoughlin, "ChatGPT: Five priorities for research," *Nature*, vol. 614, no. 9, pp. 224–226, Feb. 2023, doi: [10.1038/npp.2013.149](https://doi.org/10.1038/npp.2013.149).
- [2] E. Kasneci et al., "ChatGPT for good? On opportunities and challenges of large language models for education," *Learn. Individual Differences*, vol. 103, 2023, Art. no. 102274.
- [3] M. M. Rahman and Y. Watanobe, "ChatGPT for education and research: Opportunities, threats, and strategies," *Appl. Sci.*, vol. 13, no. 9, pp. 57–83, May 2023, doi: [10.3390/app13095783](https://doi.org/10.3390/app13095783).
- [4] K. I. Roumeliotis and N. D. Tselikas, "ChatGPT and Open-AI models: A preliminary review," *Future Internet*, vol. 15, no. 6, pp. 192–216, May 2023, doi: [10.3390/fi15060192](https://doi.org/10.3390/fi15060192).
- [5] P. Rospigliosi, "Artificial intelligence in teaching and learning: What questions should we ask of ChatGPT?," *Interact. Learn. Environ.*, vol. 31, no. 1, pp. 1–3, 2023, doi: [10.1080/10494820.2023.2180191](https://doi.org/10.1080/10494820.2023.2180191).
- [6] S. C. Kong, "A framework of curriculum design for computational thinking development in K-12 education," *J. Comput. Educ.*, vol. 3, no. 4, pp. 377–394, Nov. 2016, doi: [10.1007/s40692-016-0076-z](https://doi.org/10.1007/s40692-016-0076-z).
- [7] Ö. Korkmaz, R. Çakir, and M. Y. Özden, "A validity and reliability study of the computational thinking scales (CTS)," *Comput. Hum. Behav.*, vol. 72, pp. 558–569, Jul. 2017, doi: [10.1016/j.chb.2017.01.005](https://doi.org/10.1016/j.chb.2017.01.005).
- [8] T. C. Hsu, S. C. Chang, and Y. T. Hung, "How to learn and how to teach computational thinking: Suggestions based on a review of the literature," *Comput. Educ.*, vol. 126, pp. 296–310, Nov. 2018, doi: [10.1016/j.compedu.2018.07.004](https://doi.org/10.1016/j.compedu.2018.07.004).
- [9] S. Y. Lye and J. H. L. Koh, "Review on teaching and learning of computational thinking through programming: What is next for K-12?," *Comput. Hum. Behav.*, vol. 41, pp. 51–61, Dec. 2014, doi: [10.1016/j.chb.2014.09.012](https://doi.org/10.1016/j.chb.2014.09.012).
- [10] B. Zhong, Q. Wang, J. Chen, and Y. Li, "An exploration of three-dimensional integrated assessment for computational thinking," *J. Educ. Comput. Res.*, vol. 53, no. 4, pp. 562–590, Oct. 2015, doi: [10.1177/0735633115608444](https://doi.org/10.1177/0735633115608444).
- [11] M. Farrokhnia, S. K. Banihashem, O. Noroozi, and A. Wals, "A SWOT analysis of ChatGPT: Implications for educational practice and research," *Innov. Educ. Teach. Int.*, vol. 61, pp. 460–474, 2024, doi: [10.1080/14703297.2023.2195846](https://doi.org/10.1080/14703297.2023.2195846).
- [12] D. Mhlanga, "Open AI in education, the responsible and ethical use of ChatGPT towards lifelong learning." Feb. 11, 2023. [Online]. Available: <http://dx.doi.org/10.2139/ssrn.4354422>
- [13] J. A. González-Calero, D. Arnau, L. Puig, and M. Arevalillo-Herráez, "Intensive scaffolding in an intelligent tutoring system for the learning of algebraic word problem solving," *Brit. J. Educ. Technol.*, vol. 46, no. 6, pp. 1189–1200, Jul. 2014, doi: [10.1111/bjjet.12183](https://doi.org/10.1111/bjjet.12183).
- [14] V. J. Shute, C. Sun, and J. Asbell-Clarke, "Demystifying computational thinking," *Educ. Res. Rev.*, vol. 22, no. 1, pp. 142–158, Nov. 2017, doi: [10.1016/j.edurev.2017.09.003](https://doi.org/10.1016/j.edurev.2017.09.003).
- [15] S. S. Gill et al., "Transformative effects of ChatGPT on modern education: Emerging era of AI Chatbots," *Internet Things Cyber Phys. Syst.*, vol. 4, pp. 19–23, Jun. 2024, doi: [10.1016/j.iotcps.2023.06.002](https://doi.org/10.1016/j.iotcps.2023.06.002).
- [16] "ChatGPT," OpenAI, May 12, 2023. [Online]. Available: <https://chat.openai.com/chat05.01.2023>
- [17] P. P. Ray, "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope," *Internet Things Cyber-Phys. Syst.*, vol. 3, pp. 121–154, Apr. 2023, doi: [10.1016/j.iotcps.2023.04.003](https://doi.org/10.1016/j.iotcps.2023.04.003).
- [18] Y. Li et al., "Can large language models write reflectively," *Comput. Educ.*, vol. 4, May 2023, Art. no. 100140, doi: [10.1016/j.caeai.2023.100140](https://doi.org/10.1016/j.caeai.2023.100140).
- [19] S. Dhingra, M. Singh, S. B. Vaisakh, N. Malviya, and S. S. Gill, "Mind meets machine: Unravelling GPT-4's cognitive psychology," *Bench Council Trans. Benchmarks, Standards Eval.*, vol. 3, 2023, Art. no. 100139, doi: [10.1016/j.tbench.2023.100139](https://doi.org/10.1016/j.tbench.2023.100139).
- [20] T. H. Kung et al., "Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models," *PLoS Digit. Health*, vol. 2, no. 2, pp. 198–210, Feb. 2023, doi: [10.1371/journal.pdig.0000198](https://doi.org/10.1371/journal.pdig.0000198).
- [21] V. Handur et al., "Integrating class and laboratory with hands-on programming: Its benefits and challenges," in *Proc. IEEE 4th Int. Conf. MOOCs, Innov. Technol. Educ.*, 2016, pp. 163–168.

- [22] R. Yılmaz and F. G. K. Yılmaz, "The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation," *Comput. Educ.: Artif. Intell.*, vol. 103, 2023, Art. no. 102274, doi: [10.1016/j.caeai.2023.100147](https://doi.org/10.1016/j.caeai.2023.100147).
- [23] Y. Ramazan and Y. F. G. Karaođlan, "Examining student satisfaction with the use of smart MOOC," *Int. İstanbul Sci. Res. Congr.*, vol. 17, no. 7, pp. 264–269, Jul. 2022.
- [24] Y. Ramazan and Y. F. G. Karaođlan, "Examining student views on the use of the learning analytics dashboard of a smart MOOC," in *Proc. Int. Azerbaijan Congr. Life/Social Health Art Sci.*, 2022, pp. 82–85.
- [25] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, Mar. 2006.
- [26] A. Settle and L. Perkovic, "Computational thinking across the curriculum: A conceptual framework," Jan. 2010. [Online]. Available: <https://via.library.depaul.edu/tr/13>
- [27] International Society for Technology in Education, "Operational definition of computational thinking for k-12 education," 2011. [Online]. Available: <https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2>
- [28] Computing at School, "Computational thinking: A guide for teachers," 2015. [Online]. Available: <https://community.computingatschool.org.uk/files/8550/original.pdf>
- [29] International Society for Technology in Education, "CT Leadership Toolkit," 2015. [Online]. Available: <http://www.iste.org/docs/ctdocuments/ct-leadership-toolkit.pdf?sfvrsn=4>
- [30] D. Perez-Marín, R. Hijón-Neira, A. Babelo, and C. Pizarro, "Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children?," *Comput. Hum. Behav.*, vol. 105, Apr. 2020, Art. no. 105849, doi: [10.1016/j.chb.2018.12.027](https://doi.org/10.1016/j.chb.2018.12.027).
- [31] K. Tengler, O. Kastner-Hauler, B. Sabitzer, and Z. Lavicza, "The effect of robotics-based storytelling activities on primary school students' computational thinking," *Educ. Sci.*, vol. 12, no. 1, pp. 10–25, Jan. 2022, doi: [10.3390/educsci12010010](https://doi.org/10.3390/educsci12010010).
- [32] J. A. Lyon and A. J. Magana, "Computational thinking in higher education: A review of the literature," *Comput. Appl. Eng. Educ.*, vol. 28, no. 5, pp. 1174–1189, Sep. 2020, doi: [10.1002/cae.22295](https://doi.org/10.1002/cae.22295).
- [33] X. Tang et al., "Assessing computational thinking: A systematic review of empirical studies," *Comput. Educ.*, vol. 148, Apr. 2020, Art. no. 103798.
- [34] S. Kılıç, S. Gököđlu, and M. Öztürk, "A valid and reliable scale for developing programming-oriented computational thinking," *J. Educ. Comput. Res.*, vol. 59, no. 2, pp. 257–286, Oct. 2021, doi: [10.1177/0735633120964402](https://doi.org/10.1177/0735633120964402).
- [35] P. N. Chou, "Using ScratchJr to foster young children's computational thinking competence: A case study in a third-grade computer class," *J. Educ. Comput. Res.*, vol. 58, no. 3, pp. 570–595, Jun. 2020, doi: [10.1177/0735633119872908](https://doi.org/10.1177/0735633119872908).
- [36] B. R. Belland, A. E. Walker, N. J. Kim, and M. Lefler, "Synthesizing results from empirical research on computer-based scaffolding in STEM education: A meta-analysis," *Rev. Educ. Res.*, vol. 87, no. 2, pp. 309–344, Oct. 2017, doi: [10.3102/0034654316670999](https://doi.org/10.3102/0034654316670999).
- [37] D. J. Wood, J. S. Bruner, and G. Ross, "The role of tutoring in problem solving," *J. Child Psychiatry Psychol.*, vol. 17, no. 2, pp. 89–100, Apr. 1976, doi: [10.1111/j.1469-7610.tb00381.x](https://doi.org/10.1111/j.1469-7610.tb00381.x).
- [38] J. Melero, D. Hernández-Leo, and J. Blat, "A review of constructivist learning methods with supporting tooling in ICT higher education: Defining different types of scaffolding," *J. Univ. Comput. Sci.*, vol. 18, no. 16, pp. 2334–2360, Jan. 2012, doi: [10.3217/jucs-018-16-2334](https://doi.org/10.3217/jucs-018-16-2334).
- [39] W. Ma, O. O. Adesope, J. C. Nesbit, and Q. Liu, "Intelligent tutoring systems and learning outcomes: A meta-analysis," *J. Educ. Psychol.*, vol. 106, no. 4, pp. 901–918, Nov. 2014, doi: [10.1037/a0037123](https://doi.org/10.1037/a0037123).
- [40] R. Mermelstine, "Parent-child learning interactions: A review of the literature on scaffolding," *Brit. J. Educ. Psychol.*, vol. 87, no. 2, pp. 241–254, Feb. 2017, doi: [10.1111/bjep.12147](https://doi.org/10.1111/bjep.12147).
- [41] B. R. Belland, A. E. Walker, M. W. Olsen, and H. Leary, "A pilot meta-analysis of computer-based scaffolding in STEM education," *Educ. Technol. Soc.*, vol. 18, no. 1, pp. 183–197, Jan. 2015.
- [42] K. VanLehn, "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems," *Educ. Psychol.*, vol. 46, no. 4, pp. 197–221, Oct. 2011, doi: [10.1080/00461520.2011.611369](https://doi.org/10.1080/00461520.2011.611369).
- [43] C. Angeli and N. Valanides, "Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy," *Comput. Hum. Behav.*, vol. 105, Apr. 2020, Art. no. 105954, doi: [10.1016/j.chb.2019.03.018](https://doi.org/10.1016/j.chb.2019.03.018).
- [44] S. Basu, G. Biswas, and J. S. Kinnebrew, "Learner modeling for adaptive scaffolding in a computational thinking-based science learning environment," *User Model. User Adapt. Interact.*, vol. 27, no. 1, pp. 5–53, Jan. 2017, doi: [10.1007/s11257-017-9187-0](https://doi.org/10.1007/s11257-017-9187-0).
- [45] S. Papavaslopoulou, K. Sharma, and M. N. Giannakos, "Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences," *Comput. Hum. Behav.*, vol. 105, Apr. 2020, Art. no. 105939, doi: [10.1016/j.chb.2019.03.003](https://doi.org/10.1016/j.chb.2019.03.003).
- [46] C. Y. Wang, B. L. Gao, and S. J. Chen, "The effects of metacognitive scaffolding of project-based learning environments on students' metacognitive ability and computational thinking," *Educ. Inf. Technol.*, vol. 29, pp. 5485–5508, 2024, doi: [10.1007/s10639-023-12022-x](https://doi.org/10.1007/s10639-023-12022-x).
- [47] C. Angeli and K. Georgiou, "Investigating the effects of gender and scaffolding in developing preschool children's computational thinking during problem-solving with Bee-Bots," *Front. Educ.*, vol. 7, 2022, Art. no. 757627, doi: [10.3389/educ.2022.757627](https://doi.org/10.3389/educ.2022.757627).
- [48] X. Gao and K. F. Hew, "Toward a 5E-based flipped classroom model for teaching computational thinking in elementary school: Effects on student computational thinking and problem-solving performance," *J. Educ. Comput. Res.*, vol. 60, no. 2, pp. 512–543, Aug. 2022, doi: [10.1177/07356331211037757](https://doi.org/10.1177/07356331211037757).
- [49] X. Lai and G. K. Wong, "Collaborative versus individual problem solving in computational thinking through programming: A meta-analysis," *Brit. J. Educ. Technol.*, vol. 53, no. 1, pp. 150–170, Jan. 2022, doi: [10.1111/bjet.13157](https://doi.org/10.1111/bjet.13157).
- [50] M. Zhou and K. K. L. Lam, "Metacognitive scaffolding for online information search in K-12 and higher education settings: A systematic review," *Educ. Technol. Res. Develop.*, vol. 67, no. 6, pp. 1353–1384, Jan. 2019, doi: [10.1007/s11423-019-09646-7](https://doi.org/10.1007/s11423-019-09646-7).
- [51] A. Burgess et al., "Scaffolding medical student knowledge and skills: Team-based learning (TBL) and case-based learning (CBL)," *BMC Med. Educ.*, vol. 21, no. 1, Apr. 2021, doi: [10.1186/s12909-021-02638-3](https://doi.org/10.1186/s12909-021-02638-3).
- [52] M. Pohl, G. Wallner, and S. Kriglstein, "Using lag-sequential analysis for understanding interaction sequences in visualizations," *Int. J. Human-Comput. Stud.*, vol. 96, pp. 54–66, Dec. 2016, doi: [10.1016/j.ijhcs.2016.07.006](https://doi.org/10.1016/j.ijhcs.2016.07.006).
- [53] R. Bakeman and J. M. Gottman, "Analyzing time sequences," in *Observing Interaction: An Introduction to Sequential Analysis*, 2nd ed. Cambridge, MA, USA: Cambridge Univ. Press, 1997, pp. 150–176.
- [54] H. Kallio, A. M. Pietilä, M. Johnson, and M. Kangasniemi, "Systematic methodological review: Developing a framework for a qualitative semi-structured interview guide," *J. Adv. Nurs.*, vol. 72, no. 12, pp. 2954–2965, May 2016, doi: [10.1111/jan.13031](https://doi.org/10.1111/jan.13031).
- [55] B. Aksoy, "Coğrafya öğretiminde problem dayalı öğrenme yaklaşımı," unpublished, Institute of Education Sciences, 2004.
- [56] W. Brown, "Introduction to algorithmic thinking," 2015. [Online]. Available: www.cs4fn.com/algorithmicthinking.php
- [57] S. Psycharis and M. Kallia, "The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving," *Instruct. Sci.*, vol. 45, no. 5, pp. 583–602, Oct. 2017, doi: [10.1007/s11251-017-9421-5](https://doi.org/10.1007/s11251-017-9421-5).
- [58] G. Falloon, "An analysis of young students' thinking when completing basic coding tasks using Scratch Jr. on the iPad," *J. Comput. Assist. Learn.*, vol. 32, no. 6, pp. 576–593, Aug. 2016, doi: [10.1111/jcal.12155](https://doi.org/10.1111/jcal.12155).
- [59] L. I. González-Pérez and M. S. Ramírez-Montoya, "Components of education 4.0 in 21st century skills frameworks: Systematic review," *Sustainability*, vol. 14, no. 3, Feb. 2022, Art. no. 1493, doi: [10.3390/su14031493](https://doi.org/10.3390/su14031493).
- [60] M. Farrokhnia, Y. Baggen, H. Biemans, and O. Noroozi, "Bridging the fields of entrepreneurship and education: The role of philosophical perspectives in fostering opportunity identification," *Int. J. Manage. Educ.*, vol. 20, no. 2, Jul. 2022, Art. no. 100632, doi: [10.1016/j.ijme.2022.100632](https://doi.org/10.1016/j.ijme.2022.100632).
- [61] A. Shemshack and J. M. Spector, "A systematic literature review of personalized learning terms," *Smart Learn. Environ.*, vol. 7, no. 1, Oct. 2020, Art. no. 33, doi: [10.1186/s40561-020-00140-9](https://doi.org/10.1186/s40561-020-00140-9).
- [62] F. Ouyang, Z. Chen, M. Cheng, Z. Tang, and C. Y. Su, "Exploring the effect of three scaffoldings on the collaborative problem-solving processes in China's higher education," *Int. J. Educ. Technol. Higher Educ.*, vol. 18, no. 1, Jul. 2021, Art. no. 35, doi: [10.1186/s41239-021-00273-y](https://doi.org/10.1186/s41239-021-00273-y).
- [63] J. White et al., "A prompt pattern catalog to enhance prompt engineering with ChatGPT," 2023, *arXiv:2302.11382*.
- [64] J. Van de Pol, M. Volman, and J. Beishuizen, "Scaffolding in teacher-student interaction: A decade of research," *Educ. Psychol. Rev.*, vol. 22, no. 3, pp. 271–296, Sep. 2010, doi: [10.1007/s10648-010-9127-6](https://doi.org/10.1007/s10648-010-9127-6).



Jian Liao was born in 1978. He received the B.S. degree in computer science from Southwest University, Chongqing, China, in 2000, the master's degree in educational technology from Beijing Normal University, Beijing, China, in 2008, and the Ph.D. degree in learning, design, and technology from Penn State University, State College, PA, USA, in 2020.

He is currently an Associate Professor with the College of Education, Southwest University. His research interests include artificial-intelligence-assisted learning, teaching analysis based on computer vision, and robot-assisted language learning.



Handan Xu was born in 2003. She is currently working toward the B.Ed. degree in educational technology with the Faculty of Education, Southwest University, Chongqing, China.

Her research interests include artificial intelligence technology for education and intelligent teaching systems.



Linrong Zhong was born in 2003. She is currently working toward the B.Ed. degree in educational technology with the Faculty of Education, Southwest University, Chongqing, China.

Her research interests include education and teaching against the background of artificial intelligence.



Ming Liu was born in 1980. He received the B.S. and M.S. degrees in computer science from the University of Tasmania, Lilyfield, NSW, Australia, in 2006, and the Ph.D. degree in artificial intelligence in education from the University of Sydney, Sydney, NSW, in 2013.

From 2013 to 2014, he was a Postdoctoral Research Fellow with the School of Electrical and Information Engineering, University of Sydney. Since 2015, he has been a Lecturer, Associate Professor, and Professor with the School of Educational Technology, Southwest University, Chongqing, China. From 2018 to 2020, he was a Research Fellow at the Connected Intelligence Center, University of Technology Sydney, Sydney. He is the author of more than 40 articles. His research interests include question generation, learning analytics, writing analytics, collaborative learning analytics, and intelligent tutoring systems.

Dr. Liu was the recipient of a Ph.D. scholarship from the Australia Research Council, a postdoctoral fellowship from the Center of Research Excellence, and a research fellowship from the University of Technology Sydney. He is a Journal Reviewer for *IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES*, the *International Journal of Artificial Intelligence in Education*, and *Computers & Education*. He is a Member of the Society for Learning Analytics Research.



Longting Zhe was born in 1998. She received the B.S. degree in computer science and technology from Zhejiang International Studies University, Hangzhou, China, in 2021. She is currently working toward the master's degree in modern educational technology with the Faculty of Education, Southwest University, Chongqing, China.

Her research interests include human-computer collaborative education and educational robots.



Tao Xie was born in 1983. He received B.S. and M.S. degrees in educational technology from Southwest University, Chongqing, China, in 2007 and 2010, respectively, and the Ph.D. degree in computer science from Xi'an Jiaotong University, Xi'an, China, in 2014.

He is currently an Associate Professor with the College of Education, Southwest University. His research interests include mobile learning, educational data mining, and virtual-reality-based learning.