Exploring the Effect of Scaffolding Strategies in GenAI Chatbot on Student Engagement and Programming Skill Development

Ean Teng Khor* & Leta Chan
National Institute of Education, Nanyang Technological University, Singapore
* eanteng.khor@nie.edu.sg

Abstract: *Generative artificial intelligence (GenAI) technology has been integrated into various educational contexts since its introduction. However, research in this field often focuses more on GenAI's impact on students' learning outcomes rather than the instructional strategies employed to support learning. This study explores the effects of scaffolding strategies used by a GenAI educational chatbot on two aspects of learning: behavioral engagement with the chatbot and improvements in programming skills. 52 students interacted with different versions of a GenAI programming educational chatbot, varying in the extent of scaffolding provided. Their programming skills were assessed and compared before and after the intervention. A thematic analysis of the topics discussed in, and cognitive complexity of students' questions was also conducted. The study found that strategies which focused on enhancing conceptual understanding, as well as those that guided reflective practices, effectively fostered engagement, critical thinking, and programming skill development. These results underscore the need to align GenAI tools' functionality with students' needs to support meaningful learning. This study offers insights into the design of GenAI educational tools, opening paths for future research on training GenAI models to implement teaching pedagogies effectively.*
Keywords: Generative Artificial Intelligence, Scaffolding Strategies, Behavioral Engagement, Programming Skill Development, Learning Technologies

1. Introduction

In the 21st century, GenAI has emerged as a hot topic in learning technologies, sparking widespread efforts to integrate it into modern educational practices. In programming education, GenAI shows considerable potential by generating code and explanations of programming concepts (Bahroun et al., 2023). A review of the literature on GenAI in education by Yusuf et al. (2024) identified three research focuses: the potential benefits and risks of GenAI integration, user perceptions and experiences, and the adoption of this technology by students and educational institutions. However, there has been less emphasis on the specific pedagogical strategies GenAI employs when tutoring students and the impact of these teaching techniques on learning outcomes. To address this gap, the present study examines the effects of scaffolding strategies employed by the GenAI programming education chatbot, *MyBotBuddy* (Khor et al., 2024a), on students' learning. Specifically, it examines their behavioral engagement during learning and their performance in programming tasks.

2. Literature Review

Modern programming teaching strategies often incorporate approaches such as direct instruction, collaborative learning, situated learning, and self-directed learning (Djenic & Mitic, 2017). GenAI promises to enhance these approaches by alleviating some of the challenges faced by students and teachers, such as comprehending complex content or preparing lessons. For example, GenAI can support direct instruction by generating lesson outlines or rubrics, creating demonstration code to illustrate programming concepts, or producing visual representations that simplify complex ideas (Cooper, 2023; Liu et al., 2024). Moreover, GenAI has proven effective in generating accessible explanations of programming concepts (Lee & Song, 2024), which can empower students to engage in self-directed learning. These explanations scaffold students' understanding, equipping them to explore more advanced applications of their knowledge (Cooper, 2023). GenAI also facilitates collaborative learning by enhancing group activities such as pair programming: AI-generated reflection prompts can encourage students to share perspectives and reflect on their experiences after completing tasks (Naik et al., 2024). This process fosters cooperation and enables groups to tackle more complex programming challenges. Besides, GenAI has been employed to design programming practice exercises tailored to students' interests. Logacheva et al. (2024) found that enabling students to use GenAI to apply classroom content to real-world contexts of personal interest significantly boosted their motivation and engagement in programming learning.

However, the literature often lacks transparency regarding the teaching strategies GenAI tools use when guiding students. McGrath et al. (2024) found that most studies on GenAI chatbots in higher education made no reference to theories of educational practices. Just as teaching strategies are critical in human instruction, the strategies employed by educational GenAI tools merit equal attention. One such strategy that can be effectively implemented by GenAI tools is scaffolding. Scaffolding refers to the practice of providing guidance to help students complete tasks or understand concepts beyond their current level of expertise. Over time, the level of support is gradually reduced, empowering students to independently tackle more complex tasks or engage in higher-order thinking (Wood et al., 1976). Scaffolding has proven effective in motivating programming learners by helping them overcome initial challenges in learning a programming language, understanding problems, and devising solutions (Lin et al., 2021). Chen et al. (2024) illustrated the application of scaffolding in a GenAI coding assistant designed for elementary school students learning to code with Scratch. Recognizing the cognitive challenges younger learners face, the GenAI tool provided visual prompts to spark ideas, vivid images to represent project concepts, and a voice-guided assistant to explain coding steps, answer queries, and generate foundational code to help students progress when stuck. Students using the tool produced better code, retained more programming knowledge, and demonstrated higher engagement and motivation while learning. Similarly, Liao et al. (2024) developed a programming scaffolding system utilizing ChatGPT to enhance students' computational thinking. This system provided feedback on students' code, guided their problem-solving approaches, and addressed their questions, showcasing GenAI's potential to effectively implement scaffolding techniques to improve programming education.

Building on this foundation, the present study seeks to explore the impact of a GenAI educational chatbot on secondary school students' programming learning. Specifically, it addresses the following questions: (1) What is the impact of different scaffolding strategies used by a GenAI chatbot on students' programming ability? (2) What impact do different scaffolding strategies used by a GenAI chatbot have on students' questioning behavior when interacting with the chatbot? (3) Do students' interactions with a GenAI chatbot predict improvements in their programming abilities?

3. Research Design and Methods

A total of 60 students from four secondary schools in Singapore were recruited for this study. However, data from 8 students were excluded due to technical issues, resulting in a final sample of 52 participants (n = 52), of which six were female. The participants, aged 15 to 16 years, were students enrolled in the GCE 'O' Level Computing course. Ethics approval was obtained from the authors' institution and the Ministry of Education, along with consent from students, their parents, and their respective schools.

The study comprised three segments: a 30-minute pre-test, a one-hour intervention, and a 30-minute post-test. In the pre- and post-tests, students independently completed a Python programming task validating the check digits of ISBN-13 (pre-test) and ISBN-10 (post-test) numbers to assess their programming proficiency. During the intervention, students engaged with *MyBotBuddy* (Khor et al., 2024b), a chatbot developed based on GenAI model. Prompt engineering was leveraged in the design and development of *MyBotBuddy* which involves programming large language models through tailored prompts. The tailored prompts include breaking down a programming problem into smaller problems and giving step-by-step instruction on completing the task. The training data was preprocessed and filtered to exclude damaging or biased language. The training data sources were diversified, and bias detection and mitigation approaches were included. *MyBotBuddy*'s capabilities were enriched using a knowledge base that included the students' Computing syllabus and an API to process and handle students' requests effectively.

The students were introduced to and briefed on using *MyBotBuddy*, then encouraged to freely explore programming topics or collaborate with the chatbot to refine their pre-test task code. To examine the effects of scaffolding strategies on students' learning, *MyBotBuddy*'s feedback was continuously refined by modifying the instructional prompts provided to the chatbot to enhance its responses. Throughout the study, it underwent iterative improvements to better scaffold students' learning, resulting in four distinct versions. Each version incorporated more detailed instructions on guiding and supporting students. Table 1 highlights the changes made to improve *MyBotBuddy*'s educational impact. Each school interacted with a different version of *MyBotBuddy*, in order of recruitment.

Table 1. Iterations of MyBotBuddy

Version	*n*	Improvements
1.0 (As of Nov 2023)	12	NIL
2.0 (As of May 2024)	21	The model was instructed to ask for address students by name when responding to any queries, creating a more personalized and friendly interaction. Besides, the scope of queries it could address was expanded to include not only programming-related topics but also other relevant areas of computing, further enhancing the students' learning experience.
3.0 (As of Jul 2024)	8	The model was instructed to engage students in more dialogue by asking guiding or follow-up questions, one at a time. These questions were designed to help students better understand the nature of the programming problem they were working on.
4.0 (As of Oct 2024)	11	The chatbot's tone was adjusted to be more supportive, with a clearer focus on helping students learn programming tasks and concepts. The model was given specific goals of fostering computational thinking, critical thinking, and reflection while guiding students through programming challenges. The model was also given a framework, complete with examples, on how to encourage students to think critically, reflect on their work, debug their code, promote self-regulation in their learning, and draw connections to real-world applications. Besides, it was instructed to gradually reduce its guidance and encourage students' independence.

The first version of *MyBotBuddy*, which served as the control, was provided only basic interaction instructions. It was designed to be a helpful and friendly AI assistant, limited to computing-related enquiries. The chatbot was instructed to guide students by breaking down problems and providing step-by-step assistance without directly offering answers. Subsequent iterations incorporated increasingly detailed instructions, with the final version employing a comprehensive framework focused on promoting critical thinking, reflection, and boosting students' programming knowledge instead of simply helping students complete increasingly complicated programming tasks (Hobert, 2019). The final version was deliberately designed to allow students to utilize GenAI's generative capabilities to practice analysis and evaluation. We leveraged the cognitive domain of Bloom's taxonomy (Anderson et al., 2001) to design guiding questions that scaffolded students' critical thinking abilities, as recommended by Lim and Makany (2023). In alignment with Bloom's taxonomy, the chatbot initially focused on lower cognitive complexity levels, such as remembering and understanding, to ensure students' foundational understanding. Questions were designed to confirm students' grasp of key concepts and familiarize them with the problem at hand (De Jesus et al., 2003). For example, *MyBotBuddy* prompted students to break down programming problems into essential elements and explain their logic.

Once foundational knowledge was established, *MyBotBuddy* progressed to more advanced levels, such as analysis and application. At these levels, the chatbot prompted students to explore relationships between concepts, consider factors influencing program outcomes, and propose solutions to potential errors. *MyBotBuddy* prompted students to reflect on their problem-solving strategies and consider broader applications of their solutions. Finally, the chatbot led students towards the synthesis and evaluation levels. It encouraged them to code independently and reflect on their solutions' quality. By gradually moving from lower to higher levels of cognitive complexity, *MyBotBuddy* fostered students' critical thinking and independent learning skills alongside strengthening their foundational programming knowledge. This structured approach allowed *MyBotBuddy* to evolve from a basic assistant to an effective scaffolding educational tool aligned with established pedagogical principles.

Students' pre- and post-test tasks were graded by subject matter experts using a standardized marking rubric. The rubric awarded up to 10 points for each task for accurate application of programming functions or concepts. To evaluate the overall effectiveness of *MyBotBuddy* in enhancing students' programming performance, a paired samples *t*-test was conducted to assess significant changes in participants' pre- and post-test scores. In addition, a Kruskal-Wallis H test was used to identify any significant score differences between different iterations of the chatbot. Students' interactions with *MyBotBuddy* were recorded and transcribed for thematic analysis. The number of queries was recorded, then the topics discussed were coded as educational or non-educational. Distractions, general inquiries, and non-academic questions were considered non-educational, while educational topics included the ISBN task and questions about programming in general or other academic subjects. The cognitive complexity of the interactions was also assessed to gauge students' depth of engagement, with questions classified as either confirmation or transformation (De Jesus et al., 2003). Confirmation questions, which were further coded as remembering or understanding questions based on Bloom's taxonomy (Anderson et al., 2001), intended to promote understanding of the topic. These included requests for explanations for a concept or code, solution generation, error identification and explanation, and clarification of students' understanding of code. Transformation questions involving higher cognitive processes like experimenting or reflecting, were grouped as application, analysis, evaluation, or creating questions according to Bloom's taxonomy (Anderson et al., 2001). Questions coded under these groups included attempts to apply learned content to other contexts, experimenting with the structure of a program, evaluating the efficiency or accuracy of a program or functions, and attempts to use their existing programing knowledge and *MyBotBuddy*'s generative capabilities to create new products. Asking more transformation questions implied more cognitively complex discussions with the chatbot, which indicated deeper engagement. The frequency of each question type, the total questions asked, and the total educational engagements with the chatbot were analyzed using the Kruskal-Wallis H test and quantile regression.

4. Findings

The paired samples *t*-test revealed that students improved significantly from the pre-test (M = 6.83, SD = 3.15) to post-test (M = 7.94, SD = 2.49) following the intervention with *MyBotBuddy*, $t(51) = 2.61$, $p < .001$. GenAI educational chatbots may thus effectively enhance secondary school students' programming learning. Tests of normality and homogeneity of variance revealed that non-parametric tests were more appropriate for analysing relationships between chatbot versions, students' questioning behavior, and their score improvements. A Kruskal-Wallis H test comparing the effects of *MyBotBuddy*'s scaffolding techniques across the different iterations on students' score improvements found significant differences, $\chi 2$ (3, N = 52) = 8.96, $p = .030$. While post-hoc comparisons using Dunn's method with a Bonferroni correction for multiple tests (adjusted $p < .05$) showed no significant differences, unadjusted p-values revealed that students using the third (M = 2.75, SD = 3.33) and fourth (M = 0.91, SD = 2.21) versions of *MyBotBuddy* significantly outperformed those using the second version ($p = .020$; $p = .014$). The small sample size and conservative nature of the Bonferroni correction may have concealed potential differences. The scaffolding techniques employed in the third and fourth versions may thus have had a more pronounced effect on students' programming performance.

The Kruskal-Wallis H test was also employed to investigate differences in students' behavioral engagement with the chatbot across each version of *MyBotBuddy*. Across the four versions, no significant differences were detected in the mean number of questions asked, $\chi 2$ (3, N = 52) = 6.76, *p* = .080, the mean incidences of educational topics discussed with the chatbot, $\chi 2$ (3, N = 52) = 5.82, *p* = .121, the mean incidences of discussing the ISBN task with the chatbot, $\chi 2$ (3, N = 52) = 4.67, *p* = .197, and the mean incidences of asking remembering questions, $\chi 2$ (3, N = 52) = 6.57, *p* = .087. However, the analyses revealed a significant difference in frequency of discussion of non-educational topics across the four versions of *MyBotBuddy*, $\chi 2$ (3, N = 53) = 26.08, *p* < .001. The final group (M = 0.73, SD = 2.41) discussed non-educational topics significantly less frequently than the first (M = 9.92, SD = 19. 91, *p* = .016) and second group (M = 2.72, SD = 5.74, *p* = .000), while the third group (M = 5.5, SD = 14.38) discussed these topics significantly less often than the second group (*p* = .015). There was also a significant difference in frequency of discussion of general programming topics, $\chi 2$ (3, N = 52) = 11.97, *p* = .007, and other non-programming academic topics, $\chi 2$ (3, N = 52) = 24.89, *p* < .001, across the four versions of *MyBotBuddy*. Students interacting with the third (M = 4.5, SD = 5.32) and final versions of *MyBotBuddy* (M = 8.18, SD = 6.68) discussed general programming topics significantly more often than the second (M = 2.29, SD = 3.05, *p* = .019) group. The final group (M = 0.27, SD = 0.47) of students also discussed non-programming educational topics significantly less often than the first group of students (M = 1.58, SD = 3.42, *p* = .000), while the second group (M = 0.05, SD = 0.21) of students engaged in the same topic significantly less often than the first (p = .001) and third (M = 2, SD = 3.55, *p* = .026) group of students. A significant difference in the frequency of asking confirmation, $\chi 2$ (3, N = 52) = 10.70, *p* = .013, and transformation questions, $\chi 2$ (3, N = 52) = 12.35, p = .006, was also uncovered between the four versions of *MyBotBuddy*. Students using the final version (M = 3.63, SD = 2.06) asked significantly more confirmation questions than those using the second version (M = 2.38, SD = 2.31, *p* = .044), while students using the third version (M = 3.5, SD = 2.27) asked significantly more transformation questions than those using the first (M = 9.67, SD = 8.22, *p* = .030) and second version (M = 2.76, SD = 2.49, *p* = .004). Lastly, a significant difference was detected between the four versions of *MyBotBuddy* in the mean number of understanding, $\chi 2$ (3, N = 52) = 12.53, *p* = .006, application, $\chi 2$ (3, N = 52) = 38.24, *p* < .001, analysis, $\chi 2$ (3, N = 52) = 15.19, *p* = .002, creating, $\chi 2$ (3, N = 52) = 10.88, *p* = .012, and evaluation questions asked, $\chi 2$ (3, N = 52) = 11.35, *p* = .010. Students using the final version (M = 0.73, SD = 0.79) asked significantly more understanding questions (*p* = .003) than those using the first version (M = 3.33, SD = 2.81). Students using the second version of *MyBotBuddy* (M = 0, SD = 0) also asked significantly fewer application questions than the first (M = 1.08, SD = 2.61, *p* = .000) and fourth group of students (M = 2, SD = 1.34, *p* = .000). The third (M = 0.625, SD = 0.74) and final group of students (M = 0.27, SD = 0.65) asked significantly more analysis questions than those using the first version (M = 1.5, SD = 1.73, *p* = .003; *p* = .049). Students using the third version of *MyBotBuddy* (M = 1, SD = 1.60) also asked significantly more creating questions than the second (M = 0.62, SD = 1.28, *p* = .048) and first groups (M = 2.08, SD = 2.97, *p* = .038). Lastly, students using the third version of *MyBotBuddy* (M = 0.625, SD = 1.26) asked evaluation questions significantly more than the first (M = 5, SD = 3.74, *p* = .019) and second groups of students (M = 1.71, SD = 1.42, *p* = .014). Thus, the results indicate that the scaffolding strategies used in the third and fourth versions of *MyBotBuddy* are most successful in inducing more focused questioning behavior and a diverse range of cognitively complex questions.

The study also investigated how students' behavioral engagement with *MyBotBuddy*, measured by the total number of questions asked, the topics discussed, and the types of questions posed, influenced their post-test score improvement. To account for high heteroscedasticity and data outliers, quantile regression was employed to split students into three groups along the 25th, 50th, and 75th quantiles. This facilitated a more comprehensive analysis of the data's wide distribution, enabling us to examine differences in how students with varying levels of improvement interacted with the chatbot. At the 25th quantile, a significant negative relationship was observed between students' score improvement and their discussion of general programming topics (β = -.18, *p* = .026), suggesting that students who discussed general programming topics more frequently experienced comparatively smaller score improvement. In contrast, at the 50th quantile, score improvement was significantly positively associated with asking remembering questions (β = .28, *p*

= .019). Students who asked more remembering-related questions exhibited moderate improvements in their scores. Interestingly, a significant negative relationship was found between asking evaluation questions and students' score improvement at the 25^{th} (β = -.60, p < .001), 50^{th}, (β = -.55, p < .001), and 75^{th} quantiles (β = -.83, p = .004). This suggests that students who posed more evaluation questions generally showed smaller improvements compared to their peers at the same quantile. Thus, it is important to consider the specific type of engagement with *MyBotBuddy* to enhance learning outcomes.

5. Discussion

The study aimed to explore the effects of scaffolding techniques used by a GenAI educational chatbot, *MyBotBuddy*, on students' behavioral engagement, particularly questioning behavior, and their programming skills. The results indicated that using *MyBotBuddy* significantly improved students' scores, reinforcing the potential of GenAI chatbots to enhance students' critical thinking and programming abilities. Similarly, Yilmaz and Yilmaz (2023) observed gains in students' computational thinking, programming self-efficacy, and motivation after using ChatGPT, while Hobert (2023) demonstrated the effectiveness of a GenAI tutor that provided personalized feedback. This study offers insights into using GenAI in high school education, a less explored context, and examines student-chatbot engagement and interactions.

Comparisons across the four versions of *MyBotBuddy*, each incorporating more detailed scaffolding techniques, revealed that students who used the third and final versions benefitted most. These students were more focused and asked higher quality questions, demonstrating GenAI's effectiveness in enhancing learning through scaffolding. Students may thus benefit from responses that chunk information and include guided reflection questions. Structuring information into smaller sections reduces cognitive load to help students encode information more efficiently and retain it better (Thalman et al., 2019). This frees cognitive resources for critical evaluation and application of learned content. Later versions of *MyBotBuddy*, which emphasized supporting students' task understanding, also effectively promoted critical thinking, focus, and improved programming skills. This finding aligns with Lee and Song's (2024) finding that both students and teachers value explanations that enhance conceptual understanding, supporting this approach. Moreover, later versions of *MyBotBuddy* which included guiding and reflective questions, successfully encouraged sophisticated questioning behavior, as students asked many questions with balanced cognitive complexity. This finding contradicts concerns of complacency or overreliance due to GenAI use (Bailey, 2023), suggesting that GenAI chatbots can foster critical thinking and questioning. For teachers planning to integrate GenAI into classrooms, it may be beneficial to instruct the chatbot to break down responses into smaller parts and employ guiding and reflection questions. Students also need guidance to interact with GenAI tools appropriately, framing them as brainstorming companions instead of answer keys. Encouraging students-GenAI collaboration in classroom exercises can help instill this attitude and demonstrate its effective use.

Interestingly, the study found that engaging in more general programming discussions or asking evaluation questions was associated with smaller improvements. Selby (2015) explored an inverse relationship between the complexity of computational thinking skills and levels of Bloom's taxonomy where higher-level cognitive skills, such as evaluation, were mapped to lower-level computational thinking skills. Consequently, while students may demonstrate advanced cognitive skills, they may not possess the computational thinking skills required to break down the post-test task effectively. Scaffolding critical thinking alone may thus be insufficient to improve programming performance. GenAI programming tools could benefit from an approach that scaffolds computational thinking skills progressively, enabling students to become more well-rounded and capable of tackling complex programming problems.

6. Conclusion

Overall, the findings suggest that there is no universal solution for integrating GenAI chatbots into programming education. Simply adopting existing chatbot models may not be enough to facilitate students' learning. Instead, this study advocates a collaborative approach where educators work with GenAI to align its use with effective pedagogies that are most suited to the subject matter and students' needs. By adapting the chatbot to the learning context, educators can maximize its potential to enhance student outcomes. This study acknowledges certain limitations. The small sample size, as few schools offer GCE 'O' Level Computing, may have limited the statistical power of the analyses to detect significant group differences. Since students only interacted with *MyBotBuddy* once, many may have also approached the tool to test its capabilities, rather than allowing themselves to be guided by its prompts. Some students sought to evaluate the chatbot's response quality or how the chatbot functioned. Long-term interactions might more accurately portray how students engage with *MyBotBuddy* and its long-term effects on students' programming and critical thinking skills.

Acknowledgements

This research is supported by the Ministry of Education, Singapore, under its Academic Research Fund (RG89/22).

References

Anderson, L. W., Krathwohl, D. R. D. R., & Bloom, B. S. (2001). A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. New York: Longman.

Bahroun, Z., Anane, C., Ahmed, V., & Zacca, A. (2023). Transforming education: A comprehensive review of generative artificial intelligence in educational settings through bibliometric and content analysis. *Sustainability, 15*(17), 12983. https://doi.org/10.3390/su151712983

Bailey, J. (2023). AI in Education: The leap into a new era of machine intelligence carries risks and challenges, but also plenty of promise. *Education Next, 23*(4), 28-35.

Chen, L., Xiao, S., Chen, Y., Song, Y., Wu, R., & Sun, L. (2024). ChatScratch: An AI-augmented system toward autonomous visual programming learning for children aged 6-12. In F. F. Mueller, P. Kyburz, J. R. Williamson, C. Sas, M. L. Wilson, P. T. Dugas, & I. Shklovski (Eds.), *CHI '24: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (pp. 1-19). Association for Computing Machinery. https://doi.org/10.1145/3613904.3642229

Cooper, G. (2023). Examining science education in ChatGPT: An exploratory study of generative artificial intelligence. *Journal of Science Education and Technology, 32(3)*, 444–452. https://doi.org/10.1007/s10956-023-10039-y

De Jesus, H. P., Teixeira-Dias, J. J. C., & Watts, M. (2003). Questions of chemistry. *International Journal of Science Education, 25*(8), 1015-1034. https://doi.org/10.1080/09500690305022

Djenic, S., & Mitic, J. (2017). *Teaching strategies and methods in modern environments for learning of programming* [Paper presentation]. 14th International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2017), Algarve, Portugal. https://eric.ed.gov/?id=ED579455

Hobert, S. (2019). Say hello to 'coding tutor'! design and evaluation of a chatbot-based learning system supporting students to learn to program. *Proceedings of the International Conference on Information Systems (ICIS).* https://aisel.aisnet.org/icis2019/learning_environ/learning_environ/9

Hobert, S. (2023). Fostering skills with chatbot-based digital tutors – training programming skills in a field study. *i-com, 22*(2), 143-159. http://dx.doi.org/10.1515/icom-2022-0044

Khor, E. T., Chan, L., Koh, E., & Seow, P. (2024a). Student Perceptions of Using Generative AI Chatbot in Learning Programming. In *Proceedings of the 32nd International Conference on Computers in Education*. Asia-Pacific Society for Computers in Education.

Khor, E. T., Koh, E., & Seow, P. (2024b). Leveraging generative artificial Intelligence to support students in learning programming: A chatbot pilot study. In *Proceedings of the 18th International Conference on Learning Sciences*. International Society of the Learning Sciences.

Lee, S., & Song, K. S. (2024). Teachers' and students' perceptions of AI-generated concept explanations: Implications for integrating generative AI in computer science education. *Computers and Education: Artificial Intelligence, 7*, Article 100283. https://doi.org/10.1016/j.caeai.2024.100283

Liao, J., Zhong, L., Zhe, L., Xu, H., Liu, M., & Xie, T. (2024). Scaffolding computational thinking with ChatGPT. *IEEE Transactions on Learning Technologies, 17,* 1628-1642. https://doi.org/10.1109/TLT.2024.3392896

Lim, S. S., & Makany, T. (2023). Deploying chatbots to build students' critical thinking skills: Leveraging generative AI effectively and purposefully in higher education. In M. A. Peters, & R. Heraud (Eds.), *Encyclopedia of Educational Innovation* (pp. 1-6). Springer. http://dx.doi.org/10.1007/978-981-13-2262-4_299-1

Lin, S., Meng, N., Kafura, D., & Li, W. (2021). PDL: Scaffolding problem solving in programming courses. In C. Schulte, B. A. Becker, M. Divitini, & E. Barendsen (Eds.), *ITiCSE '21:* In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education*. Association for Computing Machinery. https://doi.org/10.1145/3430665.3456360

Liu, Z., & Guo, R., & Jiao, X., & Gao, X., & Oh, H., & Xing, W. (2024), *How AI assisted k-12 computer science education: A systematic review.* [Paper presentation] 2024 ASEE Annual Conference & Exposition, Portland, Oregon. https://peer.asee.org/47532

Logacheva, E., Hallas, A., Prather, J., Sarsa, S., & Leinonen, J. (2024). Evaluating contextually personalized programming exercises created with generative AI. In *Proceedings of the 2024 ACM Conference on International Computing Education Research*. Association for Computing Machinery. https://doi.org/10.1145/3632620.3671103

McGrath, C., Farazouli, A., & Cerratto-Pargman, T. (2024). Generative AI chatbots in higher education: A review of an emerging research area. *Higher Education,* 1-17. https://doi.org/10.1007/s10734-024-01288-w

Naik, A., Yin, J. R., Kamath, A., Ma, Q., Wu, S. T., Murray, C., Bogart, C., Sakr, M., & Rose, C. P. (2024). *Generating situated reflection triggers about alternative solution paths: A case study of generative AI for computer-supported collaborative learning*. In *Proceedings of the International Conference on Artificial Intelligence in Education*. Springer Nature Switzerland. https://link.springer.com/chapter/10.1007/978-3-031-64302-6_4

Selby, C. C. (2015). Relationships: Computational thinking, pedagogy of programming, and Bloom's Taxonomy. In *Proceedings of the Workshop in Primary and Secondary Computing Education* (pp. 80-87). Association for Computing Machinery. https://doi.org/10.1145/2818314.2818315

Thalman, M., Souza, A. S., & Oberauer, K. (2019). How does chunking help working memory?. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 45*(1), 37-55.

Wood, D. J., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry, 17*(2), 89-100. http://dx.doi.org/10.1111/j.1469-7610.1976.tb00381.x

Yilmaz, R., & Yilmaz, F. G. K. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers in Education: Artificial Intelligence, 4*, Article 100147. https://doi.org/10.1016/j.caeai.2023.100147

Yusuf, A., Pervin, N., Román-González, M., & Noor, N. M. (2024). Generative AI in education and research: A systematic mapping review. *Review of Education, 12*(2), Article e3489. https://doi.org/10.1002/rev3.3489