



Generative AI as a Teaching Assistant: A Theoretical Analysis of Its Role in Enhancing Python Programming Learning Outcomes

Ms. Nasheeda k v

Assistant Professor (contract)
computer science

P.K.K.M College of Applied Science, Mananthavady

Abstract

The fast development of artificial intelligence has dramatically changed the way education is conducted, especially in the education of programming where the conceptual complexity and diversity of learners are important challenges. This theoretical paper explores how generative AI can be used as a learning assistant to enhance the learning outcomes of Python programming. Based on well-known theories of learning like constructivism, cognitive load theory, and self-regulated learning, the paper discusses how artificial intelligence-driven tools can support customized learning, real-time feedback, and dynamism. Python is among the most commonly taught programming languages and offers a perfect setting to comprehend the pedagogical consequences of integrating AI. The paper also discusses the advantages and disadvantages of generative AI in education, such as the problems of academic integrity, over-reliance, and ethics. The results imply that generative AI has significant potential to improve programming understanding, problem-solving abilities, and learner engagement when used with consideration. The article adds to the existing debate on AI-supported learning, and offers theoretical resources to educators, researchers, and policymakers interested in using AI technologies in programming education.

Keywords: Generative AI, Python programming, learning outcomes, teaching assistant, adaptive learning, educational technology

1. Introduction

The implementation of artificial intelligence in education has brought a major paradigm shift in the teaching and learning of education, especially in technical subjects like programming (Pedro et al., 2019). The abstract concepts and logical rigor of programming education, as well as the relatively high barrier to entry, can pose special challenges to programmers in education. Python has become one of the leading programming languages thanks to its simplicity, readability, and broad applicability to various fields, including data science, machine learning, and automation. Although these features have their benefits, syntax, debugging errors, and effective use of programming logic are some of the areas that many learners find difficult to master (Thorat & Kshirsagar, 2021). In that regard, generative AI has become one of the promising tools, which can be used as a virtual teaching assistant, providing real-time assistance and individual guidance.

Generative AI is a new type of advanced machine learning that can generate human-like answers, descriptions, and solutions to user queries (Iorliam & Ingio, 2024). Such systems may be able to help learners with generating code snippets, explaining programming concepts and debugging errors and even proposing improvements. Generative AI in education is not limited to content delivery, but it can also help to create interactive and adaptive learning experiences (Farhah et al., 2025). This change is consistent with the overall trend of learner-centered education where technology is used to facilitate personalized learning. The current paper will be a theoretical discussion of how generative AI can improve the learning outcomes of Python programming through the incorporation of educational theories and technological models.

2. Conceptual Framework of Generative AI in Education

Generative AI is a field at the crossroads of artificial intelligence, natural language processing, and educational technology, thus allowing it to serve as an intelligent tutoring system (Batsaikhan & Correia, 2024). In contrast to older teaching technologies, generative AI can be actively involved in the interaction with learners, modifying its reactions, depending on the level of understanding the learner shows and his or her particular questions. This feature is consistent with the idea of intelligent tutoring systems that are designed to emulate the power of the one-on-one human tutoring. Through the use of big data and sophisticated algorithms, generative AI models have the potential to be contextually explanatory, simplifying complicated programming concepts to learners (Deriba et al., 2024).

Theoretically speaking, the application of generative AI to education can be interpreted within the context of the constructivist learning theory, which focuses on the active construction of knowledge in the course of interaction and experience (Owen, 2025). Generative AI facilitates this process by allowing learners to experiment with ideas in a conversational way, by experimenting, and through instant feedback. Also, the technology is consistent with the concept of the Zone of Proximal Development (ZPD) of Vygotsky whereby learners are able to perform tasks through guidance that they are unable to perform on their own (Rahman, 2024). Generative AI, in this context, functions as a platform, something that offers the needed assistance to close the gap between the present capability and the possible course of development. In such a way, the inclusion of generative AI into programming education is a meeting of technological advancement and educational thought.

3. Theoretical Foundations

Generative AI as a teaching assistant in Python programming can be based on a number of established learning theories (McCulloh et al., 2025). According to Sweller and Chandler (1991), Cognitive Load Theory indicates that learning works best when the instruction design reduces the extraneous cognitive load. Programming can be intrinsically cognitively demanding with its complicated syntax and problem-solving needs. Generative AI can relieve this load by decomposing complex problems into easy-to-follow steps, can give explanations, and can provide code examples. This minimizes any extraneous cognitive load and lets learners concentrate on learning fundamental concepts.

The other suitable framework is Self-regulated Learning (SRL) that emphasizes the ability of a learner to plan, monitor and review his/her learning (Johnson et al., 2019). Generative AI may assist SRL in such a way that it can give the learner instant feedback and guide him or her to identify the errors and enhance knowledge independently. Moreover, the Technology Acceptance Model (TAM) pays attention to the importance of perceived usefulness and perceptions of ease of use when it comes to the adoption of new

technologies (Davis, 1989). Generative AI is more acceptable to learners because of the ease of use and friendliness of the AI in solving problems in programming (Theresiawati et al., 2025). Putting these theoretical findings together, the generative AI within the educational realm of programming can be viewed as a motivational tool as well as a cognitive support system.

4. Role of Generative AI in Python Programming Education

Generative AI plays a complicated role in the Python programming education since it is able to address some of the most important issues that students face. It is among the principal functions that the company performs so as to assist in coding work in real-time. Students can also post their questions or code pieces and receive immediate response in form of feedback, clarification or error. This real-time feedback reduces frustrations of debugging and keeps the learners motivated. Additionally, generative AI can generate many solutions to a problem, exposing learners to many solutions and refining their abilities to solve problems. Personalization of learning experiences is the other valuable input of generative AI. Traditional classrooms do not help address the needs of diverse learners, whereas AI can support their unique learning characteristics and pace. To illustrate this, simple descriptions and the step-by-step procedure can be explained to novices and advanced students can be instructed on how to employ optimized or alternative coding procedures. Moreover, generative AI can be used to simulate interactive learning environments, where students can address problems in the form of a conversation. This interactive type style has the ability to create an improved conceptualization and understanding of programming. Overall, learning using generative AI is more interactive and dynamic.

5. Impact on Learning Outcomes

Integration of generative AI into Python programming education has an important implication to learning outcomes. Among the most significant ones is the enhancement of conceptual understanding. Generative AI can enable students to understand the basic principles of programming, including loops, functions, and data structures, by giving them clear explanations and examples of the concept in context. This improved comprehension is reflected in improved coding tasks and examination. Also, instant feedback helps the learners correct the mistakes immediately and this minimizes chances of misconceptions.

Problem-solving skills are also developed with the help of generative AI and are crucial in programming. AI tools promote creativity and analytical thinking by exposing learners to a variety of solutions and promoting experimentation. Moreover, the technology increases the interaction of the learners as learning becomes interactive and responsive. There have been positive retention rates and performance in terms of academic performance due to an increased engagement (Fredricks et al., 2004). Nevertheless, it should be stressed that the generative AI performance is determined by how well it is incorporated into the learning environment. It can be a potent tool to improve the results of programming education when applied in the right way.

6. Challenges and Limitations

Even though it has the potential to become a beneficial tool, generative AI in programming education has a number of limitations and challenges. The possibility of excessive dependence on AI tools becomes one of the primary concerns, which can impede the process of development of the ability to solve problems

independently. Students will stop using their brain and instead rely on AI-based solutions. This problem begs the question of whether assistance or autonomy in learning is the best balance.

The other problem is associated with academic integrity. The simplicity of the code formulation through the AI tools can cause some unethical activities like plagiarism or incorrectness in the exams. Teachers should thus create assessment strategies that focus on insight and creativity. Also, systems based on generative AI do not always work and can generate wrong or less-than-optimal solutions. The learners should be trained to be critical when analyzing AI-generated outputs to prevent misconceptions. Ethics, such as data privacy, and bias in AI models, should also be considered. These issues emphasize the need to use generative AI in a responsible and pedagogically reasonable way.

7. Pedagogical Implications

The inclusion of generative AI in Python programming education requires a reconsideration of teaching methods. Teachers will need to change their conventional lecture-based teaching techniques and adopt interactive and student-focused learning strategies. Generative AI may be applied as an additional instrument to supplement classroom teaching to offer more support beyond the classroom. As an example, teachers can create tasks that promote students to explore AI tools but ask them to justify their choices and responses.

Moreover, educators should be attentive to metacognitive skills of students to be able to use AI tools effectively and responsibly. This is also inclusive of training students in query writing, interpreting AI responses and verifying the quality of created code. Teacher training also needs to be a priority to ensure that teachers are ready to integrate the use of AI technologies into their activities. By aligning the teaching methodologies with technology, educational institutions can be assured of the best of generative AI and minimize the negative impacts of the technology.

8. Future Research Directions

The growth in the application of generative AI to the learning process offers numerous opportunities in terms of research opportunities in the future. They should carry out research studies to determine the effectiveness of AI-assisted learning to improve the outcome of the programming among different groups of learners. The differences between the time-proven teaching strategies and AI-based can form the basis of the comparative research, and it will provide the insight into the best practices. The longitudinal impacts of AI tools on cognitive and metacognitive processes of learners can also be investigated.

The other important area of research is the development of ethical standards of using AI in education. These include the issue of data privacy, prejudice in algorithms, and equitable access to technology. Generative AI extended to other up-and-coming technologies, such as virtual reality and gamification, is another area that researchers can research to create immersive learning classrooms. The academic community can address these gaps of research to make the use of AI in education responsible and effective.

9. Conclusion

Generative AI is an innovation in programming education, which has the ability to change the learning outcomes and address long-standing issues. By becoming a virtual teaching assistant, it provides individualized instructions, real-time feedbacks, and multimedia learning, which are in line with the existing theories of education. Generative AI could be used to improve conceptual knowledge, problem-

solving abilities and involvement of learners in the environment of Python programming. However, in order to make sure that generative AI is implemented successfully, one must be keen on its deficiencies and ethics. The middle position of teachers and schools should be moderate to adopt the power of AI and promote learning and critical thinking. As AI as a discipline is changing over the coming years, generative AI will continue to play an even bigger role in the future of education. The current theoretical discussion emphasizes that AI technologies should be implemented in the sphere of programming education carefully and evidenced-based.

References

1. Batsaikhan, B. Z., & Correia, A. P. (2024). The effects of Generative Artificial Intelligence on Intelligent Tutoring Systems in higher education: A systematic review. *Studies in Technology Enhanced Learning*, 4(1).
2. Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 13(3), 319-340.
3. Deriba, F., Sanusi, I. T., O Campbell, O., & Oyelere, S. S. (2024). Computer programming education in the age of generative AI: Insights from empirical research.
4. Farhah, N. S., Wadood, A., Alqarni, A. A., Uddin, M. I., & Aldhyani, T. H. (2025). Enhancing adaptive learning with generative ai for tailored educational support for students with disabilities. *Journal of Disability Research*, 4(3), 20250012.
5. Iorliam, A., & Ingio, J. A. (2024). A comparative analysis of generative artificial intelligence tools for natural language processing. *Journal of Computing Theories and Applications*, 1(3), 311-325.
6. Johnson, C. E., Yates, K., & Sullivan, M. E. (2019). Building a framework for self-regulated learning in surgical education: a Delphi consensus among experts in surgical education. *Journal of Surgical Education*, 76(6), e56-e65.
7. McCulloh, I., Rodriguez, P., Kumar, S., Gupta, M., Sharma, V. R., Johnson, B., & Johnson, A. N. (2025). Generative AI in Computer Science Education: Accelerating Python Learning with ChatGPT. arXiv preprint arXiv:2505.20329.
8. Owen, A. (2025). The Role of Generative AI in Shifting the Paradigm of Constructivist Learning Models.
9. Pedro, F., Subosa, M., Rivas, A., & Valverde, P. (2019). Artificial intelligence in education: Challenges and opportunities for sustainable development.
10. Rahman, L. (2024). Vygotsky's zone of proximal development of teaching and learning in STEM education. *International Journal of Engineering Research & Technology*, 13(8), 389-94.
11. Sweller, J., & Chandler, P. (1991). Evidence for cognitive load theory. *Cognition and instruction*, 8(4), 351-362.
12. Theresiawati, Hidayanto, A. N., Seta, H. B., Widyatmoko, A. M., Muhammad, M. A., Said, M. F. A., & Adifai, S. R. (2025). Analysis of factors influencing students' adoption of generative AI as a programming learning resource. *Interactive Learning Environments*, 1-18.
13. Thorat, S. A., & Kshirsagar, D. P. (2021). Developing logic building, problem solving, and debugging programming skills among students. *Journal of Engineering Education Transformations*, 402-406.