Open Access Library Journal
2025, Volume 12, e12859
ISSN Online: 2333-9721
ISSN Print: 2333-9705

Artificial Intelligence Empowered Teaching Reform and Exploration of Python Programming Course

Jinliang Liang

School of Electrical and Computer Engineering, Nanfang College Guangzhou, Guangzhou, China
Email: buptliang@hotmail.com

How to cite this paper: Liang, J.L. (2025) Artificial Intelligence Empowered Teaching Reform and Exploration of Python Programming Course. *Open Access Library Journal*, **12**: e12859.
https://doi.org/10.4236/oalib.1112859

Received: December 23, 2024
Accepted: January 24, 2025
Published: January 27, 2025

Abstract

The rapid development of Artificial Intelligence (AI) technology has profoundly impacted all sectors of society, with education and teaching undergoing particularly significant transformations driven by AI. This study examines the influence of AI on higher education, using the Python programming course at Nanfang College Guangzhou as a case study. A series of comparative experiments were conducted, in which students were divided into five groups: one using ERNIE Bot, one using Doubao, one using Kimi, one using ChatGPT, and a control group not using any AI tools. The results showed that students who used AI tools achieved a 20% - 30% increase in programming speed, demonstrated more positive and innovative thinking, and incorporated more creative elements into their projects. However, the AI tool-using groups exhibited some weaknesses in code accuracy. In contrast, the control group performed better in terms of code accuracy. Overall, the AI tool-using groups slightly outperformed the control group in terms of overall performance and achieved superior results in subject competitions. AI not only enriches teaching resources and methods but also stimulates students' innovative thinking and enhances their comprehensive literacy and independent learning abilities. However, the study also highlights that excessive reliance on AI could hinder the development of critical thinking, emphasizing the need to balance the use of AI tools with the cultivation of critical thinking in educational reform. This study provides a theoretical foundation for the teaching reform of Python programming courses and offers valuable insights for educators in developing strategies to integrate AI into teaching, aiming to improve teaching quality and better equip students for the growing demand for high-quality talent in the modern era.

Subject Areas

Software Engineering

Keywords

AI, Teaching Reform and Exploration, Python Programming Course

1. Introduction

In recent years, content-generative artificial intelligence has been rapidly developing. A range of AI applications, exemplified by ChatGPT, based on powerful deep learning algorithms and extensive data training [1], has demonstrated outstanding performance in natural language processing, code generation, and other areas. These applications can rapidly generate code frameworks for complex algorithms, explain programming concepts in detail, and provide accurate answers to various technical questions.

In the field of education, AI's influence is profound. On the one hand, it significantly expands the library of teaching materials by offering diverse case studies and real-time knowledge updates. On the other hand, it transforms the way students learn, allowing them to overcome the constraints of time and space and engage in autonomous exploration with the support of AI tools [2]. However, AI also presents challenges, such as students' over-reliance on technologies and difficulties in evaluating the accuracy and reliability of AI-generated content. In this context, investigating the effective application of AI in Python programming education is of great practical significance. This study takes Nanfang College Guangzhou as a case study for a more in-depth exploration.

Suppose the degree of students' knowledge mastery is K, the learning time is t, and the learning efficiency is η. Under the traditional teaching mode, $K = \eta t$. After the introduction of artificial intelligence, let the learning efficiency improvement coefficient be $\alpha\left(\alpha > 1\right)$, then $K = \alpha\eta t$. This formula initially indicates the potential improvement of AI on knowledge acquisition efficiency, but in fact, a variety of factors need to be comprehensively considered. For example, we can further modify the formula as $K = \alpha\eta t - \beta C$ (where C represents the negative impact caused by students' over-reliance on AI, and β is the corresponding weight coefficient), which can more comprehensively reflect the actual situation of students' learning under the influence of AI.

2. Background of Teaching Reform

The Python programming course at Nanfang College Guangzhou, originally followed a traditional teaching approach. In this method, teachers primarily lectured on theoretical concepts, grammar rules, and typical programming cases, while students practiced programming through homework assignments outside of class. While this approach helped ensure a structured understanding of the material, it inherently had limitations.

Traditional teaching resources are limited and slow to update, primarily relying on textbooks, teachers' personal experience, and the school's internal teaching platform. As a result, students were exposed to a narrow range of knowledge. When confronted with complex programming concepts and algorithms, students struggled to fully understand these topics through oral explanations and simple examples alone. In practice, students had to wait for teachers to address their questions, leading to long feedback cycles that negatively impacted student engagement and learning efficiency.

In the field of Python programming education, numerous studies have been conducted on the integration of artificial intelligence (AI) to enhance learning outcomes. For instance, in [3], Lin X. explored how intelligent tutoring systems could improve student performance by providing personalized learning paths and immediate feedback, emphasizing AI's ability to identify individual weaknesses and recommend tailored materials. Similarly, in [4], Smith *et al.* propose a model leveraging AI algorithms for personalized content recommendations, adjusting course materials based on students' learning data to facilitate better mastery of Python concepts. Furthermore, in [5], Chen *et al.* evaluate a chatbot-assisted learning environment that provides instant support and programming practice, with students reporting increased motivation and engagement through this interactive platform. Lastly, Martinez [6] presents a comprehensive review of AI technologies in Python education, discussing tools like automated grading and learning analytics that improve educational efficiency while empowering teachers. In [7], Zabala *et al.* introduce an AI-based Python programming education system that integrates a chatbot for student assistance, an automated grading system for feedback, and an entrance exam feature that suggests chapters and sections for review.

With the rise of artificial intelligence technology, modern teaching has gained new opportunities [8]. Students now have access to powerful AI tools such as ERNIE Bot (WenXinYiyan), Doubao, Kimi, and ChatGPT. These tools can generate code snippets based on students' needs, helping them understand programming logic and syntax while coding. They also offer a wide range of project cases, technical documentation, and the latest industry trends, thereby expanding learning resources. Moreover, these tools provide immediate answers to students' questions and foster creative thinking in problem-solving, enhancing students' autonomy and flexibility in learning [9]. However, while leveraging the advantages of AI, it is crucial to prevent students from becoming overly reliant on technology and ensure that they master the core knowledge and skills of programming. Achieving this balance is key to the success of teaching reform.

Despite the advantages, there is a lack of studies evaluating the impact of these AI tools on education. Therefore, it is essential to explore the role and effects of the AI tools on Python education, providing educators with valuable insights for integrating AI in teaching and understanding its broader implications for education.

3. Teaching Reform Methods

To demonstrate the impact of artificial intelligence on the Python programming course, we divided the students into two categories: those with access to artificial intelligence tools and those without. By conducting this comparative experiment, we aim to use data to illustrate the influence of AI tools on students' learning.

In the Autumn semester of 2024, 65 students majoring in Electronic Science and Technology at Nanfang College Guangzhou, enrolled in the Python Programming course. The students were divided into five groups based on their choice to use AI tools: Doubao, Kimi, ERNIE Bot, ChatGPT, and a non-AI tool group. The grouping process took into account factors such as students' programming foundation and learning abilities to ensure balance and comparability between the groups, thus enhancing the credibility of the research findings.

Based on students' preferences for AI tools, they were divided into five groups, and a voting survey was conducted. Six students abstained from voting. The results, shown in Figure 1, indicate that a larger proportion of students chose to use ERNIE Bot, accounting for 40.7%.

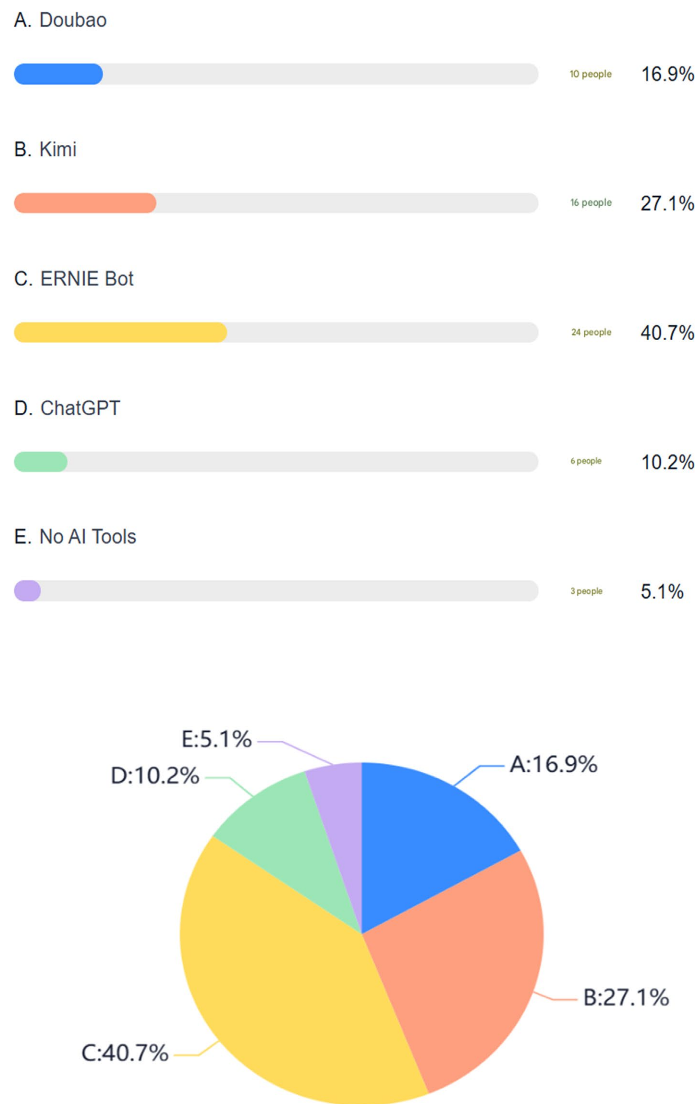

Figure 1. Groups of students using different AI tools.

3.1. Teaching Process Implementation

3.1.1. Course Content Design

The teaching is conducted in strict accordance with the standard syllabus for the Python programming course, covering core modules such as basic syntax, Python libraries, algorithm design, and project practice. In teaching basic syntax, in addition to conventional explanations, teachers guide students to use AI to explore the application scenarios of various syntax rules in different project contexts. For instance, when explaining conditional statements, students are asked to use AI to search for examples of conditional judgment in data analysis and graphical visualization, and then make comparisons [10].

For the groups using AI tools, teachers provide timely guidance to help students expand their learning of code examples with the assistance of AI [11]. For example, when studying a specific algorithm, students are instructed to request multiple algorithm implementations from AI tools and compare their advantages and disadvantages to deepen their understanding. Simultaneously, teachers guide students in using AI to explore the analysis and problem-solving approaches for complex issues. For instance, when working with image encryption algorithms, students can use AI tools to examine the differences in processing speed, accuracy, and resource consumption across various encryption algorithms [12], enabling them to select more suitable algorithms for further research and application.

For the group without access to AI tools, personalized tutoring and group collaborative learning are emphasized. Classroom interaction is increased, typical cases are explained in detail, and practical exercises are carefully designed [13]. For example, group programming competitions are organized, with teachers providing on-site guidance and addressing each group's issues.

Project-driven teaching is introduced, with a series of Python projects designed for practical application, such as data analysis systems, small game development, and intelligent customer service systems [14]. All students are required to complete these projects in groups. The groups using AI tools are encouraged to use AI appropriately to plan, generate code, and troubleshoot problems, while the non-AI group relies on close collaboration among team members and the guidance of teachers to overcome challenges in project development, thereby enhancing students' overall programming skills and fostering teamwork.

3.1.2. Learning Process Monitoring

A comprehensive learning process monitoring system is established, with regular collection of multi-dimensional data, such as code homework completion, project report quality, and classroom performance [15]. Teachers correct and comment on students' code homework weekly and provide a comprehensive evaluation of project reports, assessing aspects such as project innovation, feasibility, and the rationality of technology application.

Students using AI tools are required to document their entire process of using AI, along with their personal gains. Teachers regularly review these records and engage in discussions with students to understand their challenges and experiences.

Special attention is given to the learning progress and difficulties of the non-AI tool group, with teaching strategies adjusted flexibly. For instance, if a particular group encounters difficulties understanding a specific concept [16], additional case explanations or group discussions may be organized to deepen students' understanding.

Online learning platforms are utilized to track the learning process in real time [17], recording data such as learning time, participation in discussions, and resource downloads. This data is analyzed to identify students' learning behavior patterns, providing a basis for personalized teaching and enabling timely intervention for students exhibiting low engagement or encountering learning difficulties.

3.1.3. Teaching Method Innovation

We propose adopting the flipped classroom teaching model. Some of the teaching content is converted into micro-videos or learning materials for students to engage in autonomous learning before class, while classroom time focuses on problem discussion, project practice, and knowledge expansion. Students using AI tools are encouraged to preview relevant knowledge using AI and prepare questions for in-depth discussion with teachers and classmates during class [18], while students in the non-AI tool group preview materials through traditional methods, such as reading textbooks and watching instructional videos. In class, teachers actively guide students through group discussions, encourage them to share their learning experiences and challenges, and jointly explore solutions to promote a deeper understanding and mastery of knowledge, as well as cultivate cooperative learning and problem-solving skills.

Teaching activities that combine group cooperation learning and inquiry-based learning are implemented. During group formation, care is taken to ensure that each group includes students with different levels of AI tool usage to promote communication and learning among students [19]. In the process of project practice, group members are encouraged to leverage their individual strengths, collaborate effectively, and jointly explore the best solutions to problems. For instance, when developing a data analysis system, students using different AI tools can obtain various ideas and code snippets for data processing with the help of the tools. The group then collaborates to discuss how to integrate these resources and optimize system performance to meet project objectives [20]. Teachers play the roles of guides and supervisors, providing timely guidance and inspiration, and fostering students' teamwork and innovation skills.

The situational teaching method is applied, creating teaching scenarios closely aligned with practical applications, such as enterprise data processing requirements [21], internet application development scenarios [22], and financial data analysis contexts. Students are tasked with using Python knowledge and AI tools to solve practical problems within specific situations, thereby enhancing their learning interests and practical abilities. For example, a simulated e-commerce platform data mining scenario is set up, where students must use Python to collect and clean data, and then apply AI tools to optimize the analysis algorithms to mine valuable business insights [23], such as user purchase behavior patterns and product recommendation strategies. This approach helps students deeply understand the practical value of knowledge and stimulates their learning motivation.

3.2. Evaluation Index System Construction

3.2.1. Programming Ability Evaluation

It needs to comprehensively evaluate students' programming ability from multiple dimensions such as code writing accuracy A, efficiency E, standardization S and complex problem-solving ability C. Design a series of hierarchical and comprehensive programming test questions, including basic grammar knowledge test T_1, algorithm application ability test T_2, project development practical test T_3, etc., and conduct strict assessments in the middle and end of the course. At the same time, the test scores of each chapter are included in the evaluation of programming ability. The comprehensive score P of programming ability can be expressed as:

$$P = w_1 A + w_2 E + w_3 S + w_4 C + \sum_{i=1}^{n} w_{i+4} T_i$$

(where $w_1, w_2, w_3, w_4, \cdots, w_{n+4}$ are the weights of each dimension and test type, and $\sum_{i=1}^{n+4} w_i = 1$) We can also consider a more complex formula to evaluate programming ability. This example assumes that the complexity of the code is denoted as L,the degree of code reuse is denoted as R, and the error handling ability is denoted as H. Then the comprehensive score P of programming ability can be further expressed as:

$$P = \frac{w_1 A + w_2 E + w_3 S + w_4 C + \sum_{i=1}^{n} w_{i+4} T_i}{1 + k_1 L - k_2 R - k_3 H}$$

(where k_1, k_2, k_3 are adjustment coefficients related to code complexity, reuse and error handling ability).

3.2.2. Innovative Thinking Evaluation

We examine the innovation elements of the project, such as algorithm innovation I_a, functional module innovation I_f, solution innovation I_s, etc. Through comprehensive evaluation and scoring of project reports R, classroom presentation performance D and innovative speech in group discussions G, the score I of innovative thinking can be expressed as:

$$I = v_1 I_a + v_2 I_f + v_3 I_s + \beta_1 R + \beta_2 D + \beta_3 G$$

(where $v_1, v_2, v_3, \beta_1, \beta_2, \beta_3$ are the weights of each innovation element and evaluation link, and $v_1 + v_2 + v_3 + \beta_1 + \beta_2 + \beta_3 = 1$).

3.2.3. Learning Attitude and Autonomy Evaluation

Let's observe students' classroom participation P_a, frequency Q and quality of active questions, investment and allocation of after-class autonomous learning time T_s, etc. For the groups using AI tools, additionally evaluate their active exploration spirit E_a and self-learning management ability M of AI tools; for the non-AI tool group, focus on their self-driving force D_s and learning perseverance L. The score L_a of learning attitude and autonomy can be expressed as:

$$L_a = \lambda_1 P_a + \lambda_2 Q + \lambda_3 T_s + \lambda_4 E_a + \lambda_5 M + \lambda_6 D_s + \lambda_7 L$$

(where $\lambda_1 - \lambda_7$ are the weights of each factor).

3.2.4. Comprehensive Performance Evaluation

We comprehensively consider factors such as chapter test scores S_t, final examination scores F_t, innovative thinking I, learning attitude and autonomy L_a to calculate students' comprehensive performance G, that is:

$$G = \gamma_1 S_t + \gamma_2 F_t + \gamma_3 I + \gamma_4 L_a$$

(where $\gamma_1 - \gamma_4$ are the weights of each factor). At the same time, the award situation of subject competitions is included. The award level is set as W (first prize $W = 3$, second prize $W = 2$, third prize $W = 1$), and the number of awardees is N. Then the competition score $C = W \times N$, and the final comprehensive performance can be expressed as:

$$G' = \delta G + (1 - \delta) C$$

(where δ is the adjustment weight, $0 < \delta < 1$).

4. Analysis of Teaching Reform Results

4.1. Comparison of Programming Ability Improvement

The errors of programming can be detected by large language models AI tools [24]. And it showed in our experiments. In **Table 1**, the Improvement is code writing speed improvement ratio, and the Error is the data of programming mistakes students made. Accuracy is the accuracy Score of code, and the score is the comprehensive programming ability score students g0t.

Table 1. Comparison of different groups on various metrics.

Group	Improvement	Error	Accuracy	Score
Doubao	About 27%	5%	80	85
Kimi	About 26%	6%	78	83
ERNIE bot	About 28%	7%	75	84
ChatGPT	About 26%	8%	76	82
Non-AI tool	None	3%	90	80

The code accuracy metric was measured by testing students' code against predefined test cases. Specific criteria include the correctness of output (whether the code produced the expected results), error-free execution (no syntax or runtime errors), and efficiency (optimized time and space complexity). An automated testing system compared the output to the expected results, awarding points based on the percentage of test cases passed. Partial credit was given for solutions that passed some but not all test cases. This ensured an objective and consistent evaluation of code accuracy.

In the comprehensive programming ability test at the end of the course, the four groups using AI tools have an advantage in code writing speed. Because AI can quickly provide code framework examples, students can save time by modifying and improving them. However, in terms of code accuracy, the non-AI tool group has a lower error rate due to more strict training and guidance from teachers. Some students in the AI tool groups rely too much on AI-generated code and have a shallow understanding of code logic and weak debugging ability, resulting in some avoidable low-level errors. However, from the overall comprehensive programming ability score, the average score of the groups using AI tools is slightly higher, which indicates that AI helps to improve students' programming ability to a certain extent, but teachers need to strengthen the cultivation of students' independent judgment and code understanding ability in the teaching process.

4.2. Comparison of Innovative Thinking Stimulation

The innovative thinking metric was measured by assessing the creativity and originality of students' problem-solving approaches. Specific criteria include the novelty of the approach, such as using unique algorithms or data structures, and the integration of AI tools, with students being evaluated on how effectively they used AI tools (e.g., ChatGPT, Doubao) to enhance their solutions. Students were required to submit a reflection report explaining their approach and how AI tools were integrated, which was reviewed by instructors to assess the level of innovation. This ensured an objective evaluation of students' creativity in problem-solving.

When we accomplished the stimulation, it showed that in **Table 2**: Doubao Group leads with 9 project innovation elements and the highest evaluation score of 92, indicating strong creativity. ChatGPT Group follows closely with 8 elements and a score of 90, showing excellent innovative thinking. ERNIE Bot Group has 7 elements and a score of 88, indicating good but slightly lower performance. Kimi Group shows moderate innovation with 6 elements and a score of 85. The Non-AI Tool Group has the least with only 3 elements and a score of 75, suggesting limitations in innovation compared to AI groups. Overall, AI-related groups

Table 2. Comparison of different groups for innovation evaluation.

Group	Number of project innovation elements	Innovative thinking evaluation score
Doubao	9	92
Kimi	6	85
ERNIE bot	7	88
ChatGPT	8	90
Non-AI tool	3	75

significantly outperform the Non-AI Tool Group in both innovation elements and evaluation scores.

The groups using AI tools have more active innovative thinking in project development. For example, in a data analysis project developed by the Doubao group, through in-depth interaction with Doubao, students learned some cutting-edge data processing algorithms and visualization technologies and skillfully applied them to the project, significantly improving the innovation and practicability of the project and obtaining a high score in the innovative thinking evaluation. The ERNIE Bot group designed unique game levels and interaction modes with the inspiration of creative ideas in the game project development and performed outstandingly in the innovative thinking evaluation. In contrast, the non-AI tool group has a single information access channel, mainly relying on the cases provided by teachers and traditional teaching resources, facing great challenges in the expansion of innovative thinking and having relatively few innovation achievements.

4.3. Comparison of Changes in Learning Attitude and Autonomy

To assess the students' learning attitude and autonomy, we employ the Participation Score, Implementation Score, and Autonomous Learning as indicators to present the experimental outcomes. We compared different groups and the result is shown in Table 3. The ChatGPT Group and Doubao Group demonstrate strong initial scores and effective improvements, while the ERNIE Bot Group excels in autonomous learning time. The Non-AI Tool Group lags behind in all metrics, highlighting the advantages of AI tools in fostering innovation and learning.

Table 3. Comparison of different groups for learning attitude and autonomy.

Group	Participation score	Implementation score	Autonomous learning
Doubao	9	6	+2 hours/week
Kimi	8	5	+1.5 hours/week
ERNIE bot	8.5	5.5	+2.2 hours/week
ChatGPT	9.5	6.5	+1.9 hours/week
Non-AI tool	6	4	+0.5 hours/week

The groups using AI tools show high initial learning enthusiasm as they are curious about and eager to explore AI. They actively participate and learn, and invest more time in after-class study to improve programming skills, which significantly stimulates learning autonomy. And they excuse the learning plan better than the Non-AI group does.

4.4. Comprehensive Performance and Subject Competition Results Analysis

We evaluated students' comprehensive performance through the average score obtained in an assessment and the number of awards won in subject competitions and the results were in Table 4.

Table 4. Comparison of different groups for competition award.

Group	Score	Winners	Award levels
Doubao	90	3	First: 1, Second: 1, Third: 1
Kimi	85	2	Second: 1, Third: 1
ERNIE bot	86	2	Second: 1, Third: 1
ChatGPT	87	2	Second: 1, Third: 1
Non-AI tool	80	1	Third: 1

From the perspective of comprehensive performance, the groups using AI tools had a slightly higher average score when the students attended a programming competition in Guangdong province of China. This is mainly due to the role of AI in knowledge expansion and programming efficiency improvement. However, some students' over-reliance on AI leads to certain deductions, which offset some of the advantages. In chapter tests and final examinations, the groups using AI tools performed better in questions related to innovative thinking and knowledge application, while the non-AI tool group had more stable scores in basic knowledge and code accuracy questions. In subject competitions, the groups using AI tools have higher participation enthusiasm and more winners, indicating that AI can promote the improvement of students' innovative practical ability and competition performance. However, it is necessary to guide students in using AI correctly to avoid violations.

5. Significant Value of Artificial Intelligence in Teaching

5.1. Enriching Teaching Resources and Methods

Artificial intelligence tools provide a vast amount of rich and timely updated teaching resources for Python programming courses [25], such as diverse code examples, detailed teaching cases, and cutting-edge technical documents. Teachers can integrate these resources to make teaching content more vivid and three-dimensional. Meanwhile, the AI intelligent question-answering system can answer students' questions in real-time, and the personalized learning recommendation function can accurately push materials according to students' situations, improving the pertinence and efficiency of teaching and meeting the personalized learning needs of different students.

5.2. Stimulating Students' Innovative Ability

Through in-depth interaction with artificial intelligence, students break through the limitations of traditional teaching resources, get in touch with cutting-edge innovative knowledge, technology and concepts, broaden their thinking and knowledge boundaries, and effectively stimulate innovative thinking. In project development practice, AI tools provide inspiration and ideas at critical moments

[26], helping students break through traditional thinking patterns, try novel design schemes, and develop more innovative and competitive project works, providing strong support for cultivating students' innovative spirit and practical ability.

5.3. Cultivating Students' Information Literacy and Autonomous Learning Ability

When using AI tools, students need to screen, analyze, integrate and judge a large amount of information and discriminate the rationality and reliability of AI-generated content. This process cultivates students' information literacy, enabling them to efficiently acquire, accurately evaluate and rationally use information in the information age. Reasonable guidance for students to use AI tools for autonomous learning can exercise their autonomous learning planning [27], self-management and self-evaluation abilities, helping them develop good autonomous learning habits and laying a foundation for lifelong learning.

6. Conclusion and Outlook

The exploration and implementation of AI-powered teaching reform in the Python programming course for senior students majoring in Electronic Science and Technology at Nanfang College Guangzhou, reveals that while artificial intelligence offers significant advantages and holds substantial value in education, challenges such as students' over-reliance on AI tools and a lack of critical thinking regarding AI-generated content remain. Moving forward, it is essential for educators to refine teaching methods and strategies, ensuring students use AI tools scientifically and supervised. Emphasis should be placed on fostering autonomous learning, innovative thinking, and critical thinking skills. By leveraging the complementary strengths of AI and traditional teaching approaches, an efficient, intelligent, and innovative teaching model can be developed, facilitating the cultivation of high-quality professionals in Electronic Science and Technology who can meet the demands of the rapidly advancing technological landscape. Furthermore, ongoing attention must be paid to the evolving applications, challenges, and trends of AI in education. This will enable timely adjustments to teaching reform strategies, thereby advancing the modernization of educational practices.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] Nazir, A. and Wang, Z. (2023) A Comprehensive Survey of ChatGPT: Advancements, Applications, Prospects, and Challenges. *Meta-Radiology*, **1**, Article ID: 100022. https://doi.org/10.1016/j.metrad.2023.100022

[2] Sun, Y. and Xiao, R. (2024) Artificial Intelligence Empowering the Teaching of Ideological and Political Courses. *Teaching of Ideological and Political Courses*, **11**, 4-7.

[3] Lin, X. (2023) Application of AI in Python Programming Education: Enhancing

Learning Outcomes through Intelligent Tutoring Systems. *Journal of Educational Technology & Society*, **26**, 45-58.

[4] Smith, J. and Lee, T. (2022) Leveraging AI for Personalized Learning in Python Programming Courses. *International Journal of Artificial Intelligence in Education*, **32**, 119-137.

[5] Chen, Y. and Zhao, Q. (2023) A Chatbot-Assisted Learning Environment for Teaching Python: Design and Evaluation. *Computers & Education*, **171**, Article ID: 104226.

[6] Martinez, R. and Wang, S. (2023) Integrating AI Technologies to Improve Learning Efficiency in Python Programming: A Comprehensive Review. *Educational Technology Research and Development*, **71**, 789-814.

[7] Zabala, E. and Narman, H.S. (2024) Development and Evaluation of an Ai-Enhanced Python Programming Education System. 2024 *IEEE 15th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference* (*UEMCON*), Yorktown Heights, 17-19 October 2024, 787-792. https://doi.org/10.1109/uemcon62879.2024.10754661

[8] Huang, Y. and Deng, T. (2024) The Reform of Teacher Education and Teaching: What Should Be Done, What Is Difficult and What Can Be Done in the Era of General Artificial Intelligence. *E-Education Research*, **8**, 97-104.

[9] Tan, Z., He, L. and Chen, M. (2024) The Development Logic, Challenges and Practical Paths of Artificial Intelligence Empowering the Construction of Liberal Arts Laboratories in Colleges and Universities. *Research and Exploration in Laboratory*, **8**, 225-230.

[10] Li, J. and Zhong, L. (2024) The Application of the Artificial Intelligence Interactive Teaching Method Based on CAID in the Reform of Postgraduate Education. *Packaging Engineering*, **1**, 489-493.

[11] Yang, B., Ge, R. and Wang, Y. (2024) The Value Implication, Basic Orientation and Implementation Path of Digital and Intelligent Technologies Empowering Curriculum Reform. *Teaching in Chinese Universities*, **6**, 55-61.

[12] Li, X., Lu, J., Niu, S., Li, M. and Liu, J. (2024) Thoughts on the Construction of Basic Computer Courses and Education and Teaching in the Era of Artificial Intelligence. *Tsinghua Journal of Education*, **2**, 42-49, 70.

[13] Le, S. and Cao, M. (2023) The Era Value and Practical Approaches of Digital Empowerment in International Chinese Education. *Journal of Research on Ethnic Minority Education*, **6**, 149-156.

[14] Wu, J., Wu, F., Wen, S., Zhang, M. and Wang, J. (2023) ChatGPT Empowering Teacher Professional Development: Opportunities, Challenges and Paths. *China Educational Technology*, **5**, 15-23, 33.

[15] Cui, Z. and Ma, W. (2023) Research on the Reform of Curriculum Ideological and Political Education Empowered by Artificial Intelligence. *Theory and Practice of Education*, **12**, 33-36.

[16] Becker, B.A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J. and Santos, E.A. (2023) Programming Is Hard-Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V.* 1, Toronto, 15-18 March 2023, 500-506. https://doi.org/10.1145/3545945.3569759

[17] Chrysafiadi, K., Virvou, M., Tsihrintzis, G.A. and Hatzilygeroudis, I. (2022) Evaluating the User's Experience, Adaptivity and Learning Outcomes of a Fuzzy-Based Intelligent Tutoring System for Computer Programming for Academic Students in

Greece. *Education and Information Technologies*, **28**, 6453-6483. https://doi.org/10.1007/s10639-022-11444-3

[18] Denny, P., Kumar, V. and Giacaman, N. (2023) Conversing with Copilot: Exploring Prompt Engineering for Solving CS1 Problems Using Natural Language. *Proceedings of the* 54*th ACM Technical Symposium on Computer Science Education V*. 1, Toronto, 15-18 March 2023, 1136-1142. https://doi.org/10.1145/3545945.3569823

[19] Essel, H.B., Vlachopoulos, D., Tachie-Menson, A., Johnson, E.E. and Baah, P.K. (2022) The Impact of a Virtual Teaching Assistant (chatbot) on Students' Learning in Ghanaian Higher Education. *International Journal of Educational Technology in Higher Education*, **19**, Article No. 57. https://doi.org/10.1186/s41239-022-00362-6

[20] Hwang, G.J. and Chen, N.S. (2023) Editorial Position Paper: Exploring the Potential of Generative Artificial Intelligence in Education: Applications, Challenges, and Future Research Directions. *Educational Technology & Society*, **26**, 1-18.

[21] Kazemitabaar, M., Chow, J., Ma, C.K.T., Ericson, B.J., Weintrop, D. and Grossman, T. (2023) Studying the Effect of AI Code Generators on Supporting Novice Learners in Introductory Programming. *Proceedings of the* 2023 *CHI Conference on Human Factors in Computing Systems*, Hamburg, 23-28 April 2023, 1-23. https://doi.org/10.1145/3544548.3580919

[22] Mu, J., Fan, G., Yang, W., Feng, J., Sun, L., An, Y., *et al*. (2024) Research on Prototype Tool Improvement System Based on Power Internet Reference Application Scenario Model. 2024 *Asia-Pacific Conference on Software Engineering*, *Social Network Analysis and Intelligent Computing* (*SSAIC*), New Delhi, 10-12 January 2024, 534-538. https://doi.org/10.1109/ssaic61213.2024.00107

[23] Li, L. (2023) Analysis of E-Commerce Customers' Shopping Behavior Based on Data Mining and Machine Learning. *Soft Computing*. https://doi.org/10.1007/s00500-023-08903-5

[24] Leinonen, J., Hellas, A., Sarsa, S., Reeves, B., Denny, P., Prather, J., *et al*. (2023) Using Large Language Models to Enhance Programming Error Messages Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1, Toronto, 15-18 March 2023, 563-569. https://doi.org/10.1145/3545945.3569770

[25] Sajja, R., Sermet, Y., Cikmaz, M., Cwiertny, D. and Demir, I. (2024) Artificial Intelligence-Enabled Intelligent Assistant for Personalized and Adaptive Learning in Higher Education. *Information*, **15**, Article 596. https://doi.org/10.3390/info15100596

[26] Kayyali, M. (2024) Future Possibilities and Challenges of AI in Education. In: Kayyali, M., Ed., *Transforming Education with Generative AI*: *Prompt Engineering and Synthetic Content Creation*, IGI Global, 118-137. https://doi.org/10.4018/979-8-3693-1351-0.ch006

[27] Lin, X. (2023) Exploring the Role of ChatGPT as a Facilitator for Motivating Self-Directed Learning among Adult Learners. *Adult Learning*, **35**, 156-166. https://doi.org/10.1177/10451595231184928