ISSN 2663-8169 Vol. 7, Issue 11: 7-13, DOI: 10.25236/IJNDE.2025.071102

Research on the Innovation of Teaching Mode for Python Programming Course Driven by Artificial Intelligence

Wenbin Shao[1], Weiwei Shao[2,a*]

[1]School of Management Science and Engineering, Anhui University of Finance and Economics, Bengbu, Anhui, China
[2]School of Electronic Engineering, Tongling University, Tongling, Anhui, China
[a]2022032@tlu.edu.cn
**Corresponding author*

Abstract: *This study focuses on the application effect of the artificial intelligence-enabled blended teaching mode in Python courses and verifies its effectiveness through teaching practice. To enhance the reliability of the experiment, an experimental group and a control group were designed for comparison. The experimental group adopted a blended mode of "AI technology + traditional teaching", while the control group employed traditional teaching methods. The teaching experiment lasted 16 weeks. The research adopted a pre-test-post-test design, comprehensively assessing the impact of AI tools on students' learning interest, Python programming ability, and collaborative learning effectiveness through multi-dimensional data, including analysis of the quality of Python code written by students and surveys on students' engagement in learning Python language. The results indicated that the experimental group, adopting the "AI technology + traditional teaching" approach, significantly outperformed the control group in debugging efficiency, code complexity, and project completion (all t-values > 6.45, p-values < 0.01). The average score of the experimental group in the post-test increased by 13.3 points ($p < 0.01$), and their scores in learning interest and teaching mode satisfaction were higher than those of the control group ($p < 0.01$). Through analysis, it was found that the real-time feedback and personalized learning path design of AI tools effectively shortened the "error-correction" cycle time, reducing programming error rates by 42% and increasing students' participation in collaborative learning by 129%. The study demonstrated that the AI-enabled blended teaching mode can significantly enhance the teaching effectiveness of Python and greatly assist in improving students' programming abilities.*

Keywords: *AI; Python Programming; Blended Teaching; Teaching Mode Innovation*

1. Introduction

With the rapid development of information technology, Python, as an emerging programming language, has gradually become a core tool in fields such as artificial intelligence, web development, and data development due to its concise syntax, extensive support from third-party libraries (such as NumPy, Pandas, TensorFlow, etc.), and good cross-platform compatibility. According to the information released by the TIOBE Programming Language Index, Python has consistently ranked among the top three programming languages, and its market share is gradually expanding at this stage. Consequently, an increasing number of higher education institutions are incorporating Python as a core course for majors such as artificial intelligence and computer technology into their teaching systems. Although Python is relatively easy for students to learn, the traditional Python teaching mode still has many issues to be addressed [1]. In the teaching process, teachers often focus on teaching Python syntax rules, and students rarely have access to actual Python projects, resulting in poor practical abilities among students [2]. For instance, during the learning process, students typically only learn programming snippets from textbooks and rarely integrate the programming in textbooks with real-world application scenarios (such as writing scripts). The single evaluation model is also a drawback of the traditional Python teaching mode. Teachers usually use final exam scores as the assessment results for students, but this method finds it difficult to comprehensively evaluate students' abilities to solve practical problems. In addition, due to the limited number of teachers, they are unable to provide personalized learning guidance for every student. Students need to spend a considerable amount of time debugging Python code, leading to low

ISSN 2663-8169 Vol. 7, Issue 11: 7-13, DOI: 10.25236/IJNDE.2025.071102

learning efficiency and hindering their rapid mastery of Python-related knowledge and practical abilities [3].

In response to the current status and challenges of Python teaching, literature [4-5] indicates that Project-Based Learning (PBL) can effectively enhance students' learning interest and programming abilities. However, the implementation of PBL requires teachers to possess rich project experience and teaching resources, which imposes high demands on teachers and thus presents certain difficulties [6]. Additionally, the evaluation system of PBL needs to be improved, and how to scientifically assess students' project outcomes remains a challenge for PBL [7]. The introduction of artificial intelligence (AI) technology offers a new approach to solving these corresponding problems [8]. For example, intelligent tutoring systems can provide students with personalized project guidance, helping them quickly resolve issues encountered in projects; learning analytics platforms can record students' performance in projects, providing comprehensive evaluation bases for teachers to fully grasp students' learning situations.

With the development of artificial intelligence technology, AI has also injected new vitality into the field of education [9-10]. Utilizing some functions of AI tools, such as intelligent tutoring, automatic code generation, and learning analytics, can provide students with personalized learning paths and offer real-time feedback on programming errors and collaborative learning support [11]. For instance, GitHub Copilot can automatically generate code snippets based on code context, and CodeGeeX can analyze Python code logic in real-time and provide code optimization suggestions. Through AI technology, not only can teachers' burdens be alleviated, but students' learning experiences and outcomes can also be significantly improved.

Currently, research on the integration of AI tools into Python teaching has achieved certain progress [12-13]. Literature [14] has validated that GitHub Copilot can enhance students' programming efficiency, and literature [15] has explored how to design personalized learning paths using AI tools. Although AI technology has begun to be integrated into the teaching field and has achieved certain results, current research on integrating AI technology into teaching mostly focuses on the application of a single AI tool, lacking discussions on the long-term effects and systematic design of Python teaching [16-17]. For example, most studies only verify the effectiveness of GitHub Copilot in code generation but do not explore how to deeply integrate GitHub Copilot into classroom teaching. At present, some researchers are attempting to deeply integrate AI tools into Python teaching, but the corresponding experimental periods are short, making it difficult to assess long-term effects [18]. Currently, researchers still have certain limitations in integrating AI tools and Python teaching. How teachers can adjust the content and difficulty of Python teaching in real-time based on students' learning status and performance, and how to comprehensively evaluate students' learning abilities and practical abilities, all require further exploration by teachers [19].

Although AI tools can greatly improve students' programming efficiency, excessive reliance on AI tools may lead to the degradation of students' programming thinking. Therefore, how to correctly guide students in using AI tools and cultivate their programming and problem-solving abilities is currently an important research direction [20]. In addition, introducing AI tools into teaching also brings new challenges, such as how to design teaching plans to adapt to the characteristics of AI tools and how to assess the impact of AI tools on students' learning effects [21-22].

At present, the integration of AI technology into teaching provides new opportunities for traditional Python teaching. How to better utilize AI technology to enhance students' learning abilities and effects, and to serve Python teaching, is a problem that needs to be solved at this stage. Future teaching research should focus on the rational use of AI technology, the improvement of students' practical abilities, and the optimization of teaching plans, providing theoretical and practical guidance for the innovation of Python teaching.

2. Innovation of Python Teaching Mode Driven by Artificial Intelligence

In recent years, AI technology has developed rapidly, influencing the field of education and bringing about significant changes in Python programming teaching [23]. In traditional Python programming teaching, teachers often focus solely on explaining programming syntax, which fails to meet the needs of cultivating students' practical abilities or addressing their individual learning requirements [24]. Against this backdrop, AI technology has injected new vitality into Python teaching [25]. The following are new ideas for integrating AI technology to reform Python programming teaching.

ISSN 2663-8169 Vol. 7, Issue 11: 7-13, DOI: 10.25236/IJNDE.2025.071102

2.1 Principles of Teaching Mode Design

Personalized Learning: Every student has their own learning pace and varying areas of weakness in knowledge. AI diagnostic tools (such as Knewton) can comprehensively analyze students based on all their learning data and customize personalized learning paths based on their knowledge gaps. For example, for students who have not mastered the concept of lists, the AI system can recommend relevant teaching videos and exercises, enabling students to grasp the knowledge more quickly.

Real-time Feedback: During the process of writing Python code, students may encounter coding errors that need timely correction to avoid forming bad programming habits. AI code analysis tools (such as CodeGeeX) can monitor students' code in real-time for errors. If an error is detected, the system will promptly alert the student and provide suggestions for correction, thus optimizing the code. This real-time feedback mechanism not only enhances students' learning efficiency but also improves their error-correction abilities.

Collaborative Learning: Python teaching not only aims to equip students with good programming skills but also requires them to have a sense of cooperation. AI platforms (such as GitHub Classroom) provide a convenient environment for group project collaboration and code sharing, allowing students to jointly edit code and solve problems encountered during the learning process, thereby enhancing their team awareness and collaborative abilities.

2.2 Teaching Mode Framework

Pre-class Stage: Given the varying learning situations of each student, big data analysis is used to understand each student's learning progress and abilities, and personalized micro-lesson videos and preview tasks are recommended. Students learn before class based on the tasks recommended by the system and engage in initial programming practice. Personalized tasks can enhance students' learning interest and prepare them for related courses.

In-class Stage: After students complete their preview, AI analysis tools provide an in-depth analysis of their preview effects, allowing teachers to identify students' learning difficulties and focus on explaining these during class, enabling students to grasp the corresponding knowledge points more easily. Through group discussions and group project exercises in the classroom, students can acquire knowledge and improve their practical abilities.

Post-class Stage: After class, AI systems can grade students' assignments and automatically generate learning reports. These reports provide students with their scores, analyze their errors in detail, and offer learning suggestions and personalized learning paths to enhance their learning abilities.

2.3 Teaching Tools and Platforms

By integrating AI code generation tools with online programming platforms (such as LeetCode and Niuke.com), students can access a convenient Python programming environment. These platforms enable students to program Python online and utilize AI code generation tools for assistance, thereby facilitating better code writing.

The combination of AI data analysis tools (such as PandasAI) and virtual laboratories (such as Jupyter Notebook) also provides students with a good learning environment. After conducting experiments in the virtual laboratory, students can analyze the data provided by the system to cultivate their comprehensive abilities.

By integrating AI technology into Python teaching from multiple dimensions, not only can students' learning abilities be significantly improved, but their interest in learning Python can also be enhanced. The innovation of the Python teaching mode driven by artificial intelligence holds great significance for Python teaching.

3. Experimental Research: Verification of The Innovation of The Ai-Empowered Python Teaching Model

3.1 Experimental Design

To better verify the teaching effectiveness of the "AI + Python" blended teaching model, an

ISSN 2663-8169 Vol. 7, Issue 11: 7-13, DOI: 10.25236/IJNDE.2025.071102

experimental verification was conducted. The experiment employed a comparative experimental approach for validation. The experimental group utilized the "AI + Python" blended teaching model during instruction, while the control group continued to use the traditional Python teaching model. The Python teaching cycle lasted 16 weeks, covering content such as Python basic syntax, function modules, file operations, and structural algorithms, ensuring the breadth of the teaching scope. During instruction, learning data and performance of students in both the experimental and control groups were collected separately to deeply explore the impact of AI technology on Python teaching effectiveness.

3.2 Data Collection and Analysis

Throughout the entire experiment, to ensure the data had reference value, a multi-dimensional data collection and analysis method was adopted. Firstly, through pre-tests and post-tests, standardized programming tests were used to assess students' knowledge mastery levels. The test content included multiple dimensions such as Python basic syntax, function modules, file operations, and structural algorithms, ensuring the comprehensiveness and objectivity of the assessment results. Secondly, to better understand students' attitudes towards the introduction of AI technology in teaching, corresponding questionnaires were designed, including students' satisfaction with the teaching model, learning motivation, and learning interest. To better analyze students' attitudes, the questionnaires were presented in the form of a Likert scale. Additionally, students' code submission volume, error rates, and frequency of AI tool usage during the experiment could all be collected using code analysis tools. The relevant data records of the experiment are as follows:

Table 1 Evaluation of Programming Abilities

Indicator	Experimental Group (n=50)	Control Group (n=50)	t-value/p-value
Code Complexity (Number of Lines)	128.3 ± 25.6	95.7 ± 22.1	t=7.89, p<0.01
Debugging Efficiency (Time for Error Fixing)	12.4 ± 3.7 minutes	18.6 ± 4.2 minutes	t=8.32, p<0.01
Project Completion (Number of Tasks)	4.8 ± 0.9	3.5 ± 1.1	t=6.45, p<0.01

The t-value and p-value in Table 1 are two commonly used indicators in statistics. The t-value is the ratio of the difference between sample means to the standard error, and it can reflect the magnitude of the difference between two groups of data. The p-value is the probability of observing the current or more extreme results under the premise that the null hypothesis (H_0) holds true, and it is used to determine whether the difference is caused by random error.

Table 2 Comparison of Pre-test and Post-test Learning Outcomes

Group	Pre-experiment Learning Outcomes		Post-experiment Learning Outcomes	
	Mean Score	t-value/p-value	Mean Score	t-value/p-value
Experimental Group	62.4 ± 8.3	t=7.32, p>0.05	85.6 ± 7.2	t=9.12, p<0.01
Control Group	61.8 ± 9.1		72.3 ± 8.5	

Table 3 Learning Interest and Satisfaction (Likert 5-point Scale)

Dimension	Experimental Group	Control Group	t-value/p-value
Learning Interest	4.6 ± 0.5	3.2 ± 0.8	t=10.23, p<0.01
Satisfaction with Teaching Mode	4.8 ± 0.4	3.5 ± 0.7	t=11.45, p<0.01

Table 4 Code Quality Analysis

Group	Average number of times of using AI tools per week	Error Rate Before Experiment	Error Rate After Experiment	Error Rate Reduction
Experimental Group	12.3	28.7%	16.6%	42%
Control Group	0.5	29.1%	25.6%	12%

Table 5 Effectiveness of Collaborative Learning

Group	Average number of collaborations per person on GitHub Classroom	Average Project Score (out of 5 points)	t-value/p-value
Experimental Group	28	4.9	t=7.65, p<0.01
Control Group	12	3.8	

ISSN 2663-8169 Vol. 7, Issue 11: 7-13, DOI: 10.25236/IJNDE.2025.071102

3.3 Analysis of Experimental Results

3.3.1 Significant Improvement in Programming Ability

Data analysis from Tables 1, 2, 4, and 5 reveals that the programming ability of students in the experimental group is notably superior to that of students in the control group. Whether in terms of code complexity or debugging efficiency, the experimental results are most favorable when AI-based teaching is introduced. The experiment demonstrates that incorporating AI technology into Python instruction can effectively enhance students' programming abilities and cultivate their practical skills, enabling them to better master Python programming knowledge and achieve better teaching outcomes.

3.3.2 Notable Increase in Learning Interest

Table 3 indicates that the experimental group is more satisfied with the AI-integrated teaching model and exhibits significantly higher learning interest compared to the control group. By providing personalized teaching paths through AI technology, not only does it align with students' actual learning needs, but it also enhances their learning interest.

3.3.3 Effective Reduction in Programming Error Rates

Table 4 shows that introducing AI tools can substantially reduce programming error rates. The primary reason is that when students encounter programming errors during code writing, AI tools provide real-time reminders and optimization suggestions for code improvement. Leveraging this functionality not only enhances students' programming abilities but also fosters good programming habits.

3.3.4 High Frequency of AI Tool Usage

According to Table 4, students in the experimental group frequently use AI tools during their Python learning process. By utilizing AI tools for code generation, analysis, and modification, students' learning efficiency is improved. This phenomenon indicates that AI technology has become an indispensable learning tool in students' Python learning journey.

The experiment has validated the effectiveness of the "AI + Python" blended teaching model. Integrating AI technology into teaching not only benefits students' programming capabilities but also significantly enhances teachers' teaching efficiency. Therefore, multi-dimensional integration of AI technology into Python classrooms is an important research direction for future Python teaching.

4. Discussion and Suggestions

4.1 Discussion

Advantages of AI-Empowered Teaching Model: The AI-driven Python teaching model has brought about a revolutionary change to traditional Python teaching. By utilizing AI diagnostic tools in Python instruction, students' learning can be precisely pinpointed, and personalized learning paths can be provided. AI diagnostic tools are highly targeted and can meet the needs of different students. When students write Python code with the aid of AI code analysis tools, their coding efficiency can be significantly improved. AI code analysis tools can provide real-time error prompts and optimization suggestions for code, thereby greatly enhancing programming efficiency. Additionally, students can leverage AI platforms for collaborative learning, jointly writing and sharing code, which enhances their teamwork capabilities.

Comparison with Traditional Teaching Model: The AI-empowered Python teaching model is notably more effective than the traditional Python teaching model, significantly improving students' academic performance and learning abilities. AI platforms not only assist students in their learning but also enhance their interest in learning and increase teacher-student interaction during the teaching process.

Challenges and Solutions: Although the AI-driven Python teaching model surpasses the traditional model, it also presents certain challenges. Firstly, the transition of teachers from traditional knowledge transmitters to learning facilitators poses a significant challenge and demands higher professional competence from teachers. Secondly, excessive reliance on AI tools may have negative impacts on students, such as impairing their independent thinking abilities and hindering the cultivation of their independence. Lastly, student data privacy issues must be taken seriously, necessitating strengthened protection and management of student data.

ISSN 2663-8169 Vol. 7, Issue 11: 7-13, DOI: 10.25236/IJNDE.2025.071102

4.2 Suggestions

Enhancement of Teacher Capabilities: The introduction of AI technology has raised the bar for teachers. Therefore, schools should actively strengthen teacher training, encourage teachers to develop more AI-driven teaching resources, and involve them more in AI-driven teaching to better enhance their innovative capabilities.

Improvement of Teaching Infrastructure: Schools should provide more AI resources to students, allowing them to enjoy a wealth of educational resources on AI platforms and offering them a better learning environment.

Policy Support: Education authorities should lead the research direction of AI-empowered teaching, promote interdisciplinary collaboration, and safeguard the security and privacy of student data through legal means.

5. Conclusion and Prospects

Experiments have demonstrated that the AI-driven hybrid Python teaching approach significantly outperforms traditional Python teaching. AI technology plays a crucial role in students' personalized learning, constructing personalized learning paths, real-time code error correction, and code optimization. AI technology enhances students' interest in Python learning and their programming abilities, achieving favorable teaching outcomes.

In future teaching, more AI technologies can be introduced into Python instruction, such as integrating AI technology with virtual reality (VR) to create immersive programming environments and improve students' learning interest and experience. Teachers should also promptly update AI-driven teaching evaluation models to comprehensively assess students' programming thinking abilities and prepare for subsequent teaching reforms.

In subsequent teaching, the AI-empowered teaching model can be extended to other disciplines, promoting the application of AI technology in the teaching field and enhancing students' learning abilities and outcomes.

References

[1] Liang J .Artificial Intelligence Empowered Teaching Reform and Exploration of Python Programming Course [J].Open Access Library Journal, 2025, 12:1-14..

[2] Ambikesh J, Stasha L, Allan T, et al. Python for teaching introductory programming: A quantitative evaluation [J]. Innovation in Teaching and Learning in Information and Computer Sciences, 2011,10(1): 86-90.

[3] Sun, L, Liu J. Effects of gamified Python programming on primary school students' computational thinking skills: A differential analysis of gender [J]. Journal of Educational Computing Research,2024, 62(3): 846-874.

[4] Manuel V, Miguel A, Guillermo F, et al. A Project Based Learning Approach for Teaching Artificial Intelligence to Undergraduate Students [J]. International Journal Of Engineering Education, 2020, 36(6): 1773-1782.*

[5] Karnando, J, Muskhir M, Luthfi, A. Exploring Python Programming: A Project Based Learning-Centric Learning Experience [J]. Journal of Education Technology,2024,8(2): 306-314.

[6] Bhuyan J, Wu F, Thomas C, et al. Aerial drone: An effective tool to teach information technology and cybersecurity through project based learning to minority high school students in the US [J]. Techtrends,2020, 64 (6):899-910.

[7] Sangpom, N, Sangpom W. Project-Based Learning Combined with Inquiry-Based Learning Using Solver Tools to Promote Computational Thinking Among Undergraduate Students [J]. TEM Journal-Technology Education Management Informatics, 2025,14(1): 602-611.

[8] Philbin C A. Exploring the potential of artificial intelligence program generators in computer programming education for students [J]. ACM Inroads, 2023,14(3): 30-38.

[9] Li F, Wang L. Teaching Reform Research on Big Data Analysis and Application with Python Language [J]. International Journal of Education and Learning Systems,2022,7:55-58.

[10] Wu X, Chang J,Lian F,et al. Construction and Empirical Research of the Big Data-Based Precision Teaching Paradigm [J]. International Journal of Information and Communication Technology Education, 2022,18(2): 1-14.

ISSN 2663-8169 Vol. 7, Issue 11: 7-13, DOI: 10.25236/IJNDE.2025.071102

[11] Dafina X, Suela H, Niko H, et al. Integrating AI applications into university STEM study programs using Python [J]. International Journal of Innovative Research and Scientific Studies,2025,8(1):2374-2384.
[12] Duy, T T N, Bui A T. Application of Python for Designing Visual Aids and Integration of Artificial Intelligence (AI) to Teaching Mathematics at High Schools [J]. Journal of Science Educational Science, 2024, 69(5B): 137-148.
[13] Kehar A, Abro A, Fatima S. How Effective are AI Tools for Diverse Learners? A Case Study on Python and Data Science Education [J]. VAWKUM Transactions on Computer Sciences, 2025,13(1): 95-105.
[14] Plate D, Hutson J. Bridging Disciplines with AI-Powered Coding: Empowering Non-STEM Students to Build Advanced APIs in the Humanities [J]. ISAR Journal of Arts, Humanities and Social Sciences, 2024,9(2):44-55.
[15] Liu J, Li S. Toward artificial intelligence-human paired programming: A review of the educational applications and research on artificial intelligence code-generation tools [J]. Journal of Educational Computing Research, 2024, 62(5): 1385-1415.
[16] Namratha, M., Rekha, G, Akram S, et al. Active learning approach for python programming [J]. Journal of Engineering Education Transformations,2018 32(1): 15-19.
[17] Modran H, Ursutiu, D, Samoila, C, et al. Learning Methods Based on Artificial Intelligence in Educating Engineers for the New Jobs of the 5th Industrial Revolution [J]. Educating Engineers for Future Industrial Revolutions,2021,2: 561-571.
[18] Chen J, Ding X. Application of Artificial Intelligence in Personalized Teaching: Taking the "Python Programming Design" Course as an Example [J]. Higher Education and Practice, 2024,1(9):31-36.
[19] David R, and Mesa F. Intelligent deep-learning tutoring system to assist instructors in programming courses [J]. IEEE Transactions on Education,2023, 67(1): 153-161.
[20] Ahmed Z, Shanto S S, Jony, A I. Potentiality of generative AI tools in higher education: Evaluating ChatGPT's viability as a teaching assistant for introductory programming courses [J]. STEM Education, 2024, 4(3): 165-182.
[21] Flordeline A C. A Look Beyond the Code: Students' Perspectives on the Role of Artificial Intelligence in Transforming Python Programming Learning [J]. Seybold Report, 2024,19(11):62-69.
[22] Sun D, Boudouaia A, Zhu, C. Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study [J]. International Journal of Educational Technology in Higher Education, 2024,21(14): 1-22.
[23] Sajja, R., Sermet, Y., Cikmaz, M., et al. Artificial intelligence-enabled intelligent assistant for personalized and adaptive learning in higher education. [J] Information,2024, 15(10): 596.
[24] Davy T, Min L, Roy J Y T, et al. A review of AI teaching and learning from 2000 to 2020 [J]. Education and Information Technologies, 2023,28(7): 8445-8501.
[25] Haindl P, Gerald W. Students' experiences of using ChatGPT in an undergraduate programming course [J]. IEEE Access,2024, 12: 43519-43529.