Automation and Machine Learning (2026)
Clausius Scientific Press, Canada

DOI: 10.23977/autml.2026.070118
ISSN 2516-5003 Vol. 7 Num. 1

Research on Micro-Project-Based Python Programming Practice Teaching Reform Driven by Large AI Models

Renjie Lv, Jing Zhang

School of Information and Intelligence Engineering, University of Sanya, Sanya, China

Keywords: Large AI Models; Python Teaching; Micro-Project-Based Learning; Competency Development Model; Human–AI Collaboration

Abstract: The widespread application of large AI models has brought dual impacts on Python programming education: while enhancing learning convenience, it has also intensified students' cognitive dependence and weakened their practical programming abilities. This study constructs and implements a "micro-project-based practice teaching model driven by large AI models," decomposing course content into logically progressive micro-projects and integrating human–AI collaboration mechanisms with process-oriented evaluation. A total of 52 students participated in a one-semester teaching experiment. The results show that the average score increased from 68.5 to 79.2, the proportion of high-achieving students rose from 12% to 31%, classroom interaction rate improved from 41.5% to 78.6%, and the proportion of AI-generated code directly used decreased from 72.4% to 38.7%. Furthermore, this study establishes a competency development model and a human–AI collaboration efficiency model, with a model fitting coefficient of R2=0.87 and a 54.7% improvement in learning efficiency. The findings provide both theoretical support and practical guidance for programming education reform in the AI era, and offer a replicable framework for integrating AI technologies into foundational programming curricula.

1. Introduction

AI tools represented by large language models are profoundly reshaping programming education[1]. As a core course in general computer education, Python is characterized by a low entry threshold, strong practicality, and wide applicability[3], making it a key foundational course in artificial intelligence and data science education[4]. In recent years, the rapid advancement of generative AI technologies has accelerated the transformation of teaching paradigms, shifting from knowledge transmission to competency-oriented learning[5].

However, while AI tools significantly lower the barriers to entry, they also introduce new pedagogical challenges[2]. On the one hand, students increasingly rely on AI-generated code, often bypassing essential stages such as problem decomposition, algorithm design, and debugging[7]. This results in insufficient internalization of programming logic and weakens their computational thinking abilities[8]. On the other hand, traditional teaching models remain largely knowledge-centered, with limited integration of practice and application[9]. Issues such as fragmented project design, lack of progressive skill development, and single-dimensional evaluation further hinder effective learning outcomes[10].

Moreover, the rapid integration of AI tools into educational settings has outpaced the development of corresponding pedagogical frameworks[11]. Many instructors lack clear guidelines on how to regulate AI usage, leading to either over-restriction or uncontrolled dependence[12]. This imbalance highlights the urgent need for structured teaching models that can effectively incorporate AI while maintaining the integrity of competency development[13].

Against this background, effectively integrating AI into teaching while avoiding its negative effects has become a critical issue. This study proposes a "micro-project-based practice teaching model driven by large AI models," combining structured project design, human–AI collaboration mechanisms, and multi-dimensional evaluation strategies (Figure 1). In addition, mathematical models are introduced to quantitatively evaluate teaching effectiveness, providing a more scientific and data-driven basis for educational reform[14].

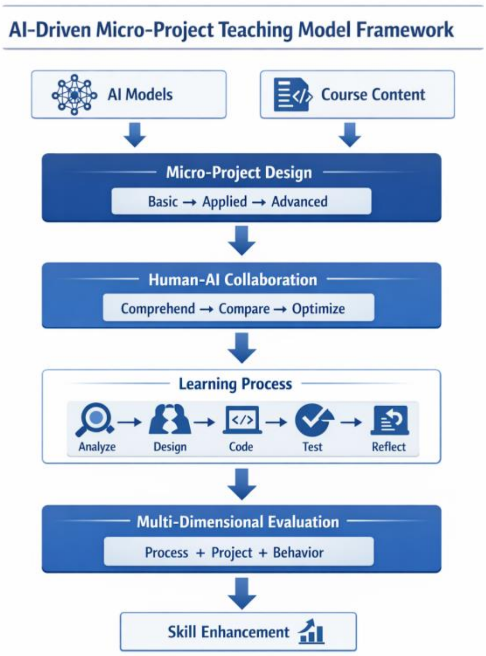

Figure 1. AI-Driven Micro-Project-Based Teaching Framework

2. Current Situation and Problem Analysis

The widespread adoption of large AI models has significantly transformed students' learning behaviors[15]. A notable trend is the emergence of "result-oriented learning," where students prioritize obtaining correct outputs rather than understanding underlying processes[6]. Many students directly use AI-generated code, skipping key cognitive steps such as logical reasoning and debugging, which leads to superficial understanding and reduced analytical ability[7].

This behavioral shift is closely related to the convenience and immediacy provided by AI tools. While such tools enhance efficiency, they also reduce the cognitive effort required for problem-solving[8]. Over time, this may lead to cognitive inertia, where students become less willing to engage in deep thinking and independent exploration[9]. As a result, their ability to transfer knowledge to new contexts is significantly weakened[10].

At the instructional level, classroom teaching still largely follows a "lecture–practice" model[9]. Exercises lack coherence and progression, making it difficult for students to build systematic knowledge structures[10]. Interaction patterns remain limited, and the introduction of AI tools further reduces students' active participation[11], as some rely on AI to complete tasks instead of engaging in problem-solving.

In terms of practical teaching, project design often lacks systematic organization[13]. Some projects are overly complex, discouraging students with weaker foundations, while others fail to align with specific knowledge points[14]. This inconsistency leads to gaps in learning progression and reduces the effectiveness of practice-based instruction. Additionally, the absence of incremental feedback mechanisms prevents students from accurately assessing their own progress[12].

Evaluation methods also exhibit significant limitations(Figure 2). Traditional approaches focus primarily on final outcomes, such as correctness and exam scores, without adequately considering learning processes or AI usage behaviors[15]. In AI-supported environments, this limitation becomes more pronounced, as it is increasingly difficult to distinguish between independently produced work and AI-assisted outputs[11]. Consequently, the validity and reliability of assessment results are compromised[13].

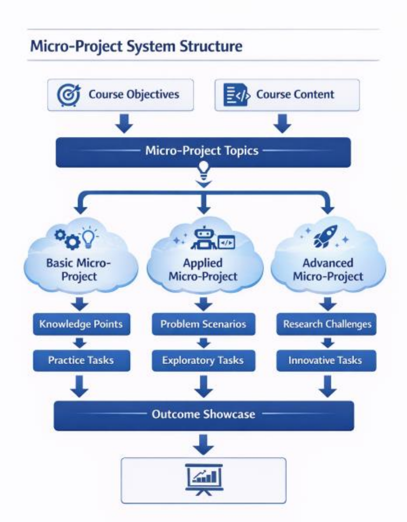

Figure 2. Hierarchical Structure of the Micro-Project System

3. Teaching Reform Design and Mathematical Model Construction

(1) Overall Design of Teaching Reform

The proposed teaching model adopts micro-projects as the fundamental instructional unit, organizing course content into three progressive levels: basic, applied, and comprehensive projects. Each micro-project targets one or two key knowledge points and requires students to complete a full cycle of analysis, design, coding, testing, and reflection.

Basic projects emphasize syntax and foundational concepts, applied projects focus on practical problem-solving in specific contexts, and comprehensive projects integrate multiple knowledge areas to develop higher-level engineering capabilities. This structured progression ensures that students gradually build both knowledge and skills while maintaining a manageable cognitive load.

In this framework, large AI models function as “cognitive partners” rather than simple tools for generating answers. Students are guided to follow a “comprehend–compare–optimize” process: first understanding AI-generated solutions, then comparing them with their own approaches, and finally refining them through critical thinking. This process encourages active engagement with AI outputs and prevents passive reliance.

Teachers play a crucial facilitative role by organizing collaborative discussions, conducting code reviews, and providing targeted feedback. Through these activities, students are encouraged to articulate their reasoning processes, identify errors, and iteratively improve their solutions. This not only enhances technical skills but also fosters metacognitive awareness.

The evaluation system is redesigned to include process evaluation, project evaluation, and behavioral evaluation. This multi-dimensional approach captures not only learning outcomes but also student engagement, problem-solving strategies, and AI usage patterns. By incorporating indicators such as code originality and AI usage rationality, the evaluation system provides a more comprehensive and accurate representation of student competencies.

(2) Competency Development Model

To quantitatively describe the cumulative effect of micro-project training, a competency development model is constructed:

$$P_i=P_0+\alpha\sum_{j=1}^{i} e^{-\beta(i-j)}\cdot(k_j)+\gamma\frac{T_{AI,i}}{T_{total,i}} \tag{1}$$

This model incorporates initial competency, project-based learning effects, and human–AI collaboration. It reflects the exponential decay of learning retention, the hierarchical nature of project difficulty, and the marginal contribution of effective AI usage. From a theoretical perspective, the model aligns with cognitive learning theories that emphasize spaced repetition and incremental reinforcement. The exponential decay term captures the diminishing influence of earlier learning experiences, while the weighting factor reflects differentiated instructional design. The inclusion of AI-related variables highlights the role of technology-mediated learning in shaping competency development.

Empirical fitting results (α=3.24, β=0.12, γ=15.6, R^2=0.87) indicate that the model effectively captures the dynamics of competency development and demonstrates strong predictive validity.

(3) Human–AI Collaboration Efficiency Model

To further analyze the impact of AI usage patterns, a learning efficiency model is proposed:

$$E=\eta\cdot\frac{S_{final}-S_{base}}{T_{total}}\cdot(1-\lambda\cdot R_{direct}) \tag{2}$$

This model demonstrates that excessive reliance on direct AI-generated code negatively affects learning efficiency, while appropriate use of AI for debugging and optimization enhances outcomes. The penalty term $\lambda\cdot R_{direct}$ quantitatively reflects the negative impact of dependency.

From an instructional design perspective, this model provides actionable insights. It suggests that teaching strategies should aim to reduce direct AI dependency while encouraging exploratory and assistive uses of AI. By regulating AI interaction patterns, educators can optimize learning efficiency and promote deeper engagement.

Experimental data confirm a significant improvement in efficiency after reform, highlighting the importance of guided AI usage. The model also provides a quantitative basis for designing adaptive teaching interventions.

4. Teaching Implementation and Effect Analysis

(1) Teaching Implementation

The teaching experiment was conducted over a 16-week semester with 52 undergraduate students enrolled in the course Fundamentals of Artificial Intelligence and Python Programming. The instructional design integrated micro-project-driven learning, structured human–AI collaboration, and multi-dimensional evaluation mechanisms, supported by an online learning platform for continuous data tracking and behavioral analysis.

During the implementation process, the “loop structures” module was selected as a representative case to illustrate the effectiveness of the teaching model. Before class, students were required to complete structured preparatory tasks, including conceptual review and simple coding exercises. As a result, the pre-class completion rate reached 84.6%, indicating a substantial improvement in learning initiative.

In-class activities were organized around a “random lottery program” micro-project. The instructor first introduced a real-world problem scenario, guiding students to decompose the task into manageable components. Students then collaboratively designed algorithms and implemented solutions, with the instructor providing real-time feedback. Compared with traditional teaching, classroom interaction increased significantly, and students demonstrated higher levels of engagement and participation.

After class, students were assigned an extended task requiring them to enhance the original program by incorporating additional features such as probabilistic simulation, statistical analysis, and visualization. This task encouraged students to apply knowledge in more complex contexts and to explore multiple solution paths. The completion rate reached 91.3%, and many students voluntarily implemented advanced optimizations, reflecting increased motivation and autonomy.

(2) Effect Analysis

The results of the teaching experiment indicate substantial improvements across multiple dimensions. In terms of academic performance, students’ average scores increased significantly, accompanied by a notable rise in the proportion of high-achieving students and a marked decrease in failure rates. The reduction in score dispersion suggests that the teaching model not only improved overall performance but also reduced individual differences.

From the perspective of learning behavior, students exhibited stronger engagement throughout the learning process. Classroom interaction rates nearly doubled, and weekly study time increased considerably. More importantly, students demonstrated a more balanced distribution of learning efforts across pre-class preparation, in-class participation, and post-class consolidation, indicating the formation of more sustainable learning habits.

In terms of programming competency, improvements were observed in multiple aspects, including code quality, functional completeness, and logical accuracy. The frequency of logical errors decreased significantly, while the use of structured coding practices and annotations increased. These changes indicate a transition from surface-level coding to more systematic and disciplined programming practices.

A particularly important finding is the transformation of AI usage behavior. The proportion of students relying on direct code generation decreased substantially, while the use of AI for debugging, optimization, and conceptual understanding increased. This shift demonstrates that students gradually moved from passive dependence to active collaboration with AI tools. At the same time, code originality improved significantly, reflecting enhanced independent thinking and creativity.

Survey data further corroborate these findings. Most students expressed positive attitudes toward the micro-project-based teaching model and reported improvements in their practical skills. A large proportion also indicated that the structured guidance on AI usage helped them develop more responsible and effective learning strategies. These results suggest that the proposed model not only improves performance but also fosters long-term learning competencies.

These behavioral changes are visually summarized in the bar chart *Learning Behavior and AI Usage Comparison*, which quantifies key indicators before and after the teaching reform. As shown in the figure, classroom interaction percentage rose from 42% to 79%, and average weekly study time increased from around 3.5 hours to 6 hours, reflecting enhanced learning engagement. Pre-class preparation rate climbed from 49% to 85%, validating the improvement in students' proactive learning initiative. For AI usage patterns, the share of direct AI-assisted code generation dropped sharply from 73% to 39%, while the proportion of using AI for debugging and optimization surged from 28% to 62%, fully verifying the transformation from passive AI dependence to active human-AI collaboration. Finally, code originality increased from 55% to 79%, further confirming the enhancement of students' independent thinking and creative coding abilities. This chart provides intuitive, data-driven evidence for the effectiveness of the teaching model in optimizing learning behaviors and guiding responsible AI usage(Figure 3).

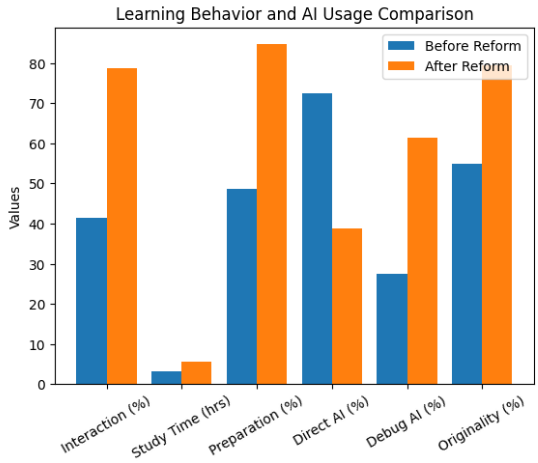

Figure 3. Learning Behavior and AI Usage Comparison

5. Contributions and Innovations

5.1 Model Innovation

This study proposes a competency development model and a human–AI collaboration efficiency model, providing quantitative tools for evaluating AI-driven teaching reforms and offering a new perspective for analyzing learning processes in AI-supported environments.

5.2 Mechanism Innovation

A "comprehend–compare–optimize" human–AI collaboration framework is established, promoting deeper cognitive processing and critical thinking while effectively mitigating the negative effects of technological dependence.

5.3 Evaluation Innovation

A multi-dimensional evaluation system incorporating learning processes, project performance, and AI usage behavior is developed, significantly improving the validity and reliability of assessment in AI-integrated learning environments.

5.4 Design Innovation

A three-level progressive micro-project system is designed, enabling the integration of knowledge acquisition and competency development through incremental, high-frequency learning tasks, thereby enhancing learning continuity, engagement, and achievement.

6. Conclusions and Future Work

This study constructs and validates a micro-project-based Python programming teaching model driven by large AI models. Both experimental results and mathematical analysis demonstrate its effectiveness in improving academic performance, learning behaviors, programming competencies, and AI usage patterns.

The competency development model reveals the cumulative effects of project-based learning and the marginal benefits of human–AI collaboration, while the efficiency model quantitatively explains how AI usage patterns influence learning outcomes. Together, these models provide a scientific basis for optimizing teaching strategies in AI-enhanced environments.

Despite these contributions, several limitations remain. Accurately monitoring AI usage behavior remains technically challenging, and further refinement of differentiated teaching strategies is needed. Additionally, the sample size and experimental duration are limited, which may affect the generalizability of the findings.

Future research will focus on developing intelligent learning analytics systems, exploring deeper cognitive mechanisms of human–AI collaboration, and extending the proposed model to other courses and educational contexts. Furthermore, integrating adaptive learning technologies and personalized feedback mechanisms may further enhance the effectiveness of the teaching model.

Acknowledgement

This work was supported by The Research on Collaborative Prediction Model of Financial Market Volatility in Hainan Free Trade Port Based on Multi-source Heterogeneous Data (Hnky2026-38)

References

[1] Luckin, R., Holmes, W., Griffiths, M., & Forcier, L. B. (2016). Intelligence unleashed: An argument for AI in education. Pearson.
[2] Holmes, W., Bialik, M., & Fadel, C. (2019). Artificial intelligence in education: Promises and implications for teaching and learning. Center for Curriculum Redesign.
[3] Van Rossum, G., & Drake, F. L. (2009). Python 3 reference manual. CreateSpace.
[4] Guzdial, M. (2015). Learner-centered design of computing education. Morgan & Claypool.
[5] Zawacki-Richter, O., et al. (2019). Systematic review of research on artificial intelligence applications in higher education. International Journal of Educational Technology in Higher Education, 16(1), 39.
[6] Kasneci, E., et al. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, 103, 102274.
[7] Finnie-Ansley, J., et al. (2022). The robots are coming: Exploring the implications of OpenAI Codex on introductory programming. ACM Conference on International Computing Education Research.
[8] Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33–35.
[9] Biggs, J., & Tang, C. (2011). Teaching for quality learning at university. McGraw-Hill.
[10] Prince, M. (2004). Does active learning work? A review of the research. Journal of Engineering Education, 93(3), 223–231.
[11] Cotton, D. R. E., Cotton, P. A., & Shipway, J. R. (2023). Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. Innovations in Education and Teaching International.
[12] Aiken, R. M. (1979). Intelligent tutoring systems. Artificial Intelligence Review, 3(4), 251–277.
[13] Merrill, M. D. (2002). First principles of instruction. Educational Technology Research and Development, 50(3), 43–59.
[14] Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(3), e1355.
[15] Becker, S. A., et al. (2017). NMC horizon report: 2017 higher education edition. EDUCAUSE.