Advances in Educational Technology and Psychology (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/aetp.2025.090615
ISSN 2371-9400 Vol. 9 Num. 6

Research on Reshaping College Students' Learning Motivation and Teaching Pathways in the AI Era-Taking Python Course as an Example

Yan Ding

Guangzhou College of Technology and Business, Guangzhou, Guangdong, 510850, China

Keywords: Artificial Intelligence; Higher Education; Learning Motivation; Human-Machine Symbiosis; Teaching Reform; Python Teaching

Abstract: Nowadays, the rapid development of artificial intelligence has brought unprecedented opportunities and challenges to higher education. University students generally believe that "knowledge mastered by AI is equivalent to their own mastery," which leads to educational dilemmas such as cognitive laziness and low learning motivation. Based on constructivist learning theory, self-determination theory, and human-machine collaboration concepts, this paper focuses on analyzing the psychological mechanisms behind university students' misconceptions about artificial intelligence and constructs a "human-machine symbiotic" learning model. The research is conducted from four aspects: role positioning reconstruction, curriculum teaching innovation, core competency cultivation, and teacher role transformation.

1. Introduction

The rapid development of artificial intelligence has profoundly impacted higher education. According to 2025 statistics from the Ministry of Education, 621 universities in China have established artificial intelligence majors, with over eighty-five percent of courses involving artificial intelligence[1]. Python, as one of the most popular programming languages, has become a compulsory general education course in many universities. However, this technological penetration has a dual impact. A notion that "what AI already knows is equivalent to our own mastery" is spreading among university students, especially in programming learning. Students feel that they only need to send prompts to AI to generate code and then copy and paste it to complete tasks, severely affecting their enthusiasm for learning programming and leading to problems such as insufficient cultivation of programming thinking and declining debugging skills.

A tracking study by the Tsinghua University Institute of Education showed that students using AI for learning assistance initially had slight advantages, but after half a month, the learning outcomes of AI-dependent students significantly declined, with periodic exam results far inferior to those of self-learning students[2]. This survey result illustrates the potential damage of over-reliance on AI to deep learning capabilities. A teaching experiment at MIT found that when students were allowed to use AI tools to solve Python programming assignments, although they could complete assignments quickly, their final exam results were significantly worse than those of previous similar student cohorts[3].

Amid the wave of AI-assisted teaching, how to guide university students in the rational application of AI tools and harmonious coexistence with artificial intelligence is an unavoidable issue in Python language teaching. Based on educational paradigm transformation theory and human-machine collaborative learning concepts, this paper analyzes the characteristics of higher education in the AI era and university students' learning psychology, combined with teaching cases from Python courses, to construct a systematic set of teaching guidance strategies aimed at addressing the lack of learning motivation among university students due to the popularization of AI technology and promoting the deep integration of AI and education.

2. Theoretical Basis and Literature Review

2.1 Educational Paradigm Transformation in the AI Era

The impact of AI on education is essentially a revolutionary transformation of the learning paradigm. In 2024, Jiang Guohua proposed that learning in the AI era requires two fundamental transformations: the first is from "traditional knowledge-instillation learning" to "modern competency-oriented learning," with the core shift from merely pursuing knowledge memorization to focusing on knowledge application and basic competency cultivation; the second is the "collaborative advancement of knowledge acquisition and competency generation" catalyzed by the AI era, emphasizing the efficient integration of information using AI tools to break through learning barriers while focusing on the deep generation of core literacies such as critical thinking and innovation capabilities[4].

Python, as the programming language for artificial intelligence, undergoes even greater changes. Previous Python teaching focused on syntax explanation and simple program code writing. In the AI era, Python teaching should shift its focus to the cultivation of computational thinking abilities and the enhancement of complex problem-solving skills. Students must learn to coexist harmoniously with AI, treating AI as a collaborative partner in programming, not merely a code-generating machine.

2.2 AI Education Application from the Perspective of Self-Determination Theory

Self-determination theory posits that humans have three basic psychological needs: autonomy, competence, and relatedness. When these three needs are supported and satisfied, they can effectively promote the formation and development of intrinsic motivation[5]. Under the influence of artificial intelligence, the core transformation in Python language teaching must focus on these three needs. There are significant differences between traditional Python teaching and AI-era Python teaching, as shown in Table 1. AI supports students' needs for competence, autonomy, and relatedness through mechanisms such as constructing personalized learning paths, providing adaptive challenges, and intelligently matching learning communities.

However, everything has two sides. If artificial intelligence is not correctly applied, it can inhibit students' psychological needs, severely dampen their learning motivation. In 2024, literature by Google Education Vice President Shantanu Sinha pointed out that university students are using artificial intelligence in "creative ways," but these ways often exceed teachers' current cognitive frameworks. Simply categorizing students' use of AI as "cheating" may be a misunderstanding, as students themselves realize that "without their own thinking and debugging, merely copying and pasting code by sending prompts to AI is useless for improving their abilities and may even harm their learning capabilities"[6].

Table 1: Comparison between Traditional Python Teaching and AI-Era Python Teaching

Comparison Dimension	Traditional Python Teaching	AI-Era Python Teaching
Core Objective	Syntax mastery and simple programming	Computational thinking and problem-solving ability
Teacher-Student Relationship	Teacher-centered code demonstration	Student-centered inquiry learning
Technology Role	Auxiliary debugging tool	Cognitive partner and code reviewer
Core Competency	Memorizing syntax, understanding logic	Critical thinking, algorithm design, debugging ability
Evaluation Method	Code correctness test	Diversified, process-oriented evaluation

3. Analysis of University Students' AI Cognitive Misconceptions and Current Situation

3.1 Manifestations of AI Cognitive Misconceptions in Python Learning

Current university students' misconceptions about AI seriously affect their learning motivation. They generally believe that AI can provide perfect answers through prompts, so many students feel it is unnecessary to learn, thinking that learning is useless. From the perspective of Python course assignments, many students over-rely on AI code generation, directly copying and pasting AI-generated code for their assignments. They treat tools like ChatGPT and Deepseek as code generators. When receiving assignments, they consider how to use prompts to make AI generate suitable code rather than thinking about how to solve practical problems themselves. Jiang Guohua clearly pointed out in a general basic course learning guidance lecture that this over-reliance on AI behavior weakens independent thinking ability and leads to the degradation of deep learning capabilities[4].

Secondly, university students do not realize the importance of cultivating programming thinking in the Python course. Many students believe that using AI to generate correct Python code based on prompts is sufficient, but core abilities such as algorithm design for problems, problem decomposition, and code debugging techniques cannot be replaced by AI. Teacher Fu Qiang emphasized that AI is an extension of cognition, not a replacement, and College students should transform from "passive storers of grammatical rules" into "proactive commanders steering the ship of intelligence."[7].

3.2 Internal Mechanism of Declining Python Learning Motivation

Students believe that problems encountered in learning and even future work can be solved simply by sending prompts to AI. This idea reflects a lack of learning motivation, belonging to the phenomenon of external motivation substitution pointed out in learning motivation theory. Originally, it was difficult for students to write large codes and experiment reports, but now they can easily complete them by sending a prompt, with higher quality. Students feel there is no need to study hard, lacking external motivation to learn more difficult content. However, this cognitive misconception overlooks the intrinsic value of programming language teaching-the programming learning process itself has an irreplaceable role in developing computational thinking and cultivating problem-solving abilities.

From an educational psychology perspective, the decline in students' learning motivation mainly stems from self-efficacy imbalance, goal orientation deviation, and lack of instant gratification preference. Self-efficacy imbalance refers to students believing under the influence of AI that no matter how hard they try, their programming ability cannot compare to AI, so there's no need to study hard with AI available; goal orientation deviation refers to students only focusing on code running results while ignoring the thinking construction process required for problem-solving during programming; lack of instant gratification preference refers to students expecting to quickly obtain executable code during the learning process while avoiding in-depth exploration. As Google Sinha pointed out[8], students actually realize that completely relying on AI to complete programming tasks is not beneficial for personal ability improvement because they know this practice may ultimately harm their learning ability and academic performance. This view indicates that students still have a need for deep learning and genuine ability cultivation deep down, with the key being how teachers take effective measures to provide correct guidance. University students' cognitive misconceptions about AI and their impact on Python learning are mainly reflected in the following four aspects, as shown in Table 2:

Table 2: University Students' Cognitive Misconceptions about AI and Their Impact on Python Learning

Cognitive Misconception	Main Manifestation	Impact on Python Learning
Tool Substitution Theory	Believing AI can completely replace human programming	Decreased learning motivation, degraded debugging ability
Infinite Capability Theory	Overestimating AI's code ability, ignoring its limitations	Weakened critical thinking, over-reliance on generated code
Learning Simplification Theory	Simplifying programming to code acquisition	Neglecting algorithm thinking and problem decomposition ability training
Ethical Neutrality Theory	Ignoring copyright and ethical issues of AI code	Increased academic misconduct risk, weakened professional ethical awareness

4. Practical Pathways and Strategies for Educational Guidance

4.1 Role Positioning Reconstruction: From Code Receiver to Intelligent Navigator

In summary, to directly address the problem of insufficient motivation in students' Python learning process, it is necessary to guide students in establishing a scientific understanding of human-machine collaboration. Teacher Fu Qiang once proposed that students should transition from passive "knowledge receivers" to active "intelligent navigators" capable of applying technology, an insight highly enlightening[7]. Course instructors should help students recognize that artificial intelligence can generate code, but AI is not a shortcut to replace human thinking; rather, it is a collaborative partner that expands human cognitive boundaries and enhances thinking depth.

In the Python language teaching process, this role transformation should be reconstructed from three aspects: cognition, ability, and practice:

First, regarding cognition, guide students to abandon the misconception of viewing AI as our programming competitor and instead build a new cognitive framework of "AI as a programming microscope and telescope." Course instructors can demonstrate through specific examples how to use AI to assist students in understanding complex algorithms, while emphasizing that humans always play an irreplaceable core role in problem definition and algorithm selection when solving complex problems.

Second, regarding ability cultivation, focus on cultivating four types of literacy in students: first, critical thinking and precise problem definition ability, enabling students to pose high-quality prompts to AI; second, interdisciplinary integration and innovative application ability, promoting students to design practically meaningful Python solutions combined with their professional backgrounds; third, humanized expression ability, focusing on improving code readability and user experience; fourth, value judgment and ethical decision-making ability, guiding students to correctly identify potential ethical impacts and social consequences of AI-generated code.

Thirdly, in terms of practice, a learning approach that utilizes artificial intelligence as a teaching assistant should be implemented: Firstly, students should use artificial intelligence for pre-class preview and self-study of new course content; Secondly, students apply AI to solve the corresponding exercises in this chapter, allowing AI to assist in providing multiple problem-solving methods, thereby choosing the most appropriate one. Thirdly, students use artificial intelligence to debug code and analyze the causes of errors. Fourth, students utilize AI-assisted functions to expand, continuously optimize program performance, broaden their thinking, and enhance their abilities.

4.2 Curriculum and Teaching Method Innovation

Python teaching reform in the AI era must be implemented in specific curriculum reconstruction and teaching method innovation. Xi'an Jiaotong University's "Digital Learning Assistance" practice provides a good reference. They comprehensively construct a digital teaching environment, fully upgrade intelligent teaching platforms, and explore AI-empowered course learning in all elements, making students "active"[8]. Based on this, the following Python language teaching methods are designed:

4.2.1 "AI Dual-Teacher" Collaborative Teaching Model

Introduce the "AI dual-teacher" collaborative teaching model in Python teaching, where AI appears as a teaching assistant role, with the teacher leading, forming a "human-machine collaboration" teaching closed loop. The course teacher leads concept explanation and thinking method elaboration, while the AI teaching assistant is responsible for personalized learning materials, code error analysis, and code modification suggestion push, achieving optimal allocation of teaching resources and further optimizing students' learning experience, enabling them to learn higher-quality knowledge in the same amount of time.

Specific implementation paths include the following four links:

In the concept explanation link, the teacher not only explains the rules themselves but focuses on explaining the core syntactic elements of Python and deeply analyzing the computational thinking logic and program design concepts behind them.

In the AI-assisted exercise link, the teacher carefully analyzes student learning conditions to organize personalized programming assignments. AI dynamically adjusts question difficulty through learning condition diagnosis, achieving tailored education, arranging simpler exercises for students with poorer foundations and designing challenging questions for students with better foundations.

In the code review link, the teacher teaches students to use AI for code review, conducting real-time multi-dimensional analysis of the code, including detecting syntax correctness, code style, execution efficiency, readability, etc., from programming perspectives to provide optimization suggestions, helping students establish a good engineering specification awareness.

In the reflection and iteration link, the teacher guides students to further modify and debug code based on the code modification suggestions fed back by the AI teaching assistant, but must carefully record specific modification ideas and learning insights. This process not only improves students' programming ability but also helps cultivate metacognitive ability, prompting students to shift from passive knowledge acceptance to active knowledge system construction.

4.2.2 Problem-Oriented AI Collaborative Project Learning

It is necessary to analyze students' learning situations and design a problem-driven, AI-assisted, and collaborative project-based learning model for them to lay a solid foundation. Teachers should focus on enhancing students' ability to solve practical problems and instruct them on how to correctly and efficiently utilize artificial intelligence tools. This teaching model emphasizes the systematic and reflective nature of the project process, with the following specific implementation steps:

First, in the problem definition stage, students choose project themes with practical significance in groups, and each group conducts discussions to clarify project goals and functional requirements. During this process, the teacher guides students in task decomposition and preliminary definition of AI usage boundaries, thinking about which links are suitable for introducing AI assistance and which parts require independent design and implementation, to cultivate students' teamwork, task planning, and AI assistance evaluation abilities.

Second, in the AI-assisted system design stage, students can use AI tools to participate in system architecture design and algorithm construction, but submitted assignments must record detailed experimental processes, explaining the decision basis and technology selection reasons for each design link, thereby strengthening students' systematic thinking and logical expression abilities.

Third, in the segmented development mechanism stage, the teacher divides the project development process into multiple stages, clearly stipulating which links in the assignment can use AI assistance and which links must be completed by oneself, to ensure students' core programming abilities and algorithmic thinking are fully trained.

Fourth, in the AI code review and feedback response stage, set up AI code review links at key project nodes, with AI conducting multi-dimensional evaluation of submitted code. Students revise the code based on review feedback and write written explanations, elaborating specific reasons for accepting or rejecting AI suggestions, thereby enhancing their code review ability and critical thinking.

Fifth, in the project defense and learning reflection stage, require students to focus on demonstrating the specific application methods of AI in various project stages, the role it played, and their own learning gains and ability growth during the collaboration process in the final project defense, promoting students' transformation from "tool use" to "cognitive construction" for deep-level learning.

4.2.3 Implementing Interdisciplinary Integrated Python Project Design

To deepen the application value of artificial intelligence tools and stimulate students' intrinsic learning motivation, drawing on the "cross-boundary learning" concept, an interdisciplinary integrated Python project module is designed. This module aims to break the knowledge barriers between computer science and other professional fields, guiding students to experience the instrumental value of Python programming and AI technology while solving real problems in their professional fields, achieving a cognitive leap from "learning programming" to "using programming thinking to solve problems in their own field."

Specific implementation paths are as follows:

Professional scenario-oriented interdisciplinary project design: Design personalized projects closely related to students' majors according to the characteristics of students from different professions, keeping up with the forefront of professional development. Setting assignments with actual utility can closely connect programming learning with students' core knowledge systems, effectively promoting cognitive transfer and integrated innovation.

Comparative research and application strategy study of diversified AI tools: Encourage students to actively try and compare multiple AI tools during project development, analyzing the responses, advantages, and limitations of at least three AI tools for the same problem. Students need to describe the AI tool selection strategy and usage evaluation clearly in the project report, thereby cultivating their technology selection ability and critical use awareness of AI tools.

Supporting knowledge grid construction for cross-boundary integration: Guide students to actively construct an "interdisciplinary knowledge tree" connecting computer science and their own majors through extensive literature reading, case studies, and project practice. In this training process, students not only need to master Python's technical syntax but also understand its application paradigms and problem-solving modes in their professional fields, increasing the challenge compared to traditional course learning requirements, thus forming a T-shaped knowledge structure capable of flexibly mobilizing various knowledge to solve complex problems, laying the foundation for becoming interdisciplinary composite innovative talents in the future.

4.3 Focus and Shift in Core Competency Cultivation

In the context of deep involvement of artificial intelligence technology in the educational process, the goal of Python teaching must achieve a strategic shift from "skill impartation" to "literacy cultivation." The core lies in focusing on cultivating those high-level human cognitive and value judgment abilities that are difficult for artificial intelligence to simulate or replace. The following four types of abilities should become the core anchors of Python teaching in the AI era:

(1) Critical thinking and precise problem definition ability

With the empowerment of generative AI in the code implementation phase, students' ability to precisely define and analyze problems becomes particularly important. Therefore, the focus of our teaching needs to shift from "how to code" to "why code with this strategy." Students must learn to pose structurally clear, role-clear, style-clear, boundary-distinct, and logically rigorous complex prompts to AI tools and be able to critically examine and evaluate multiple solutions fed back by AI. For example, in data analysis projects, one should not vaguely require "analyze data" but need to clearly define analysis objectives, data preprocessing processes, key indicators, and expected visualization output forms.

(2) Interdisciplinary integration and contextualized innovative application ability

The vitality of Python lies in its powerful integration ability as a "glue language" in diverse scenarios. Teaching design should consciously guide students to combine programming skills with their research professional fields, catalyzing interdisciplinary innovative solutions, further enhancing students' knowledge accumulation. This integrated practice not only deepens professional learning but also shapes students' cross-boundary innovative thinking.

(3) Humanized design and communication ability

In the current situation where AI can generate functional code, writing clear, readable, maintainable code that focuses on the end-user experience becomes a core advantage of human programmers. This "humanized ability" is concretely manifested as clear variable naming, detailed comment documentation, modular structure design, and meticulous optimization of user interaction interfaces. Cultivating this ability essentially cultivates students' engineering literacy, collaborative spirit, and user empathy.

(4) Value judgment and ethical decision-making ability

Facing potential ethical risks such as copyright disputes, privacy leaks, and algorithmic biases that may lurk in AI-generated code, cultivating students' value judgment and ethical decision-making ability has become indispensable. Teaching should introduce relevant cases to guide students to examine and discuss the social impact of their programming practices, establishing a sense of responsible innovation.

The cultivation of the above abilities needs to be deeply integrated into teaching models such as project-based learning and case studies, and students are encouraged to participate in broader technical practice communities. By practicing in real and complex task contexts, students can achieve comprehensive improvement of core literacy.

4.4 Teacher Role Transformation and Professional Ability Development Path

In the context of deep integration of artificial intelligence and education, Python course teachers urgently need to achieve role transformation from "knowledge transmitters" to "learning experience designers" and "computational thinking coaches." This transformation requires teachers not only to master new technologies but also to reconstruct their teaching philosophies and methodologies. The "Reform Framework for Higher Education Quality Improvement" proposes a systematic teacher development system covering three dimensions: "technological literacy-teaching methods-interdisciplinary collaboration"[9]. Based on this framework, specific paths for teacher ability improvement are as follows:

4.4.1 Technological Literacy Dimension: Implementing the "AI+Python Teaching Empowerment Plan"

To achieve AI-assisted teaching, teachers' application level of artificial intelligence must be improved, so the "AI+Python Teaching Empowerment Plan" must be implemented. All Python language course teachers should proficiently master various AI tools, skillfully use AI for code analysis, automatic assessment, generate personalized learning materials, and integrate the use of these tools into the classroom, combining AI to complete course design, task implementation, and learning evaluation. Through empowerment, teachers can use technological means to gain insight into students' thinking processes and learning difficulties, thereby achieving precise teaching intervention, rather than staying at the level of simple tool operation.

4.4.2 Teaching Method Dimension: Deepening the "Human-Machine Dual-Track Collaboration" Teaching Model

Teachers should be encouraged to skillfully apply and implement the "human-machine dual-track collaboration" teaching model. The roles and responsibilities of teachers and artificial intelligence in the teaching process should be accurately defined: As the core leaders of teaching, teachers are responsible for inspiring learners, guiding higher-order thinking, promoting collaborative inquiry, and providing humanistic care. Artificial intelligence, on the other hand, is a powerful assistant, undertaking tasks such as personalized content delivery, real-time feedback, and data-driven learning and analysis. Professional development training should focus on helping teachers design programming tasks that encourage deep thinking among students and guide them in establishing constructive learning partnerships with AI.

Xi'an Jiaotong University's "Digital Teaching Assistant" practice provides a good reference for this. The university effectively made classroom teaching "active" by systematically constructing high-quality teaching resources, developing high-quality digital textbooks, and comprehensively promoting blended teaching reform[8]. This successful experience shows that teachers' roles are shifting from traditional "knowledge transmitters" to learning designers promoting student "ability construction."

4.4.3 Interdisciplinary Collaboration Dimension: Building a "School-Enterprise-Research" Professional Development Community

To break through the disconnection between Python teaching and professional application, efforts should be made to build an open, collaborative "school-enterprise-research" professional development community. By establishing inter-school teacher development centers, school-enterprise joint laboratories, and school-research cooperation projects, promote deep communication and collaboration between Python lead teachers and experts from various disciplinary fields, industry engineers, and scientific researchers. The goal is to jointly develop teaching cases and practical projects originating from real scenarios, enabling Python teaching to truly take root in interdisciplinary problem soil, helping students grow into composite talents with the ability to solve complex problems.

5. Reconstruction of Evaluation System and Safeguard Mechanisms

5.1 Multidimensional Evaluation Indicator Design

The traditional evaluation system centered on code correctness can no longer meet the needs of cultivating innovative Python talents in the AI era, and a multi-dimensional evaluation mechanism must be constructed. The "Reform Framework for Higher Education Quality Improvement" proposes a multi-dimensional evaluation system covering technological innovation, social contribution value, industry alignment, etc.[9], which can be concretized in Python courses as:

Technological innovation dimension: Introduce indicators such as code innovation, algorithm efficiency, and solution uniqueness, not limited to correctness.

Social responsibility dimension: Consider code readability, maintainability, user experience, and social impact.

Industry alignment dimension: Invite enterprise engineers to participate in project evaluation, focusing on the engineering quality and practical application value of the code.

5.2 Learning Process Evaluation Reform

Under the influence of artificial intelligence, grading standards need to pay more attention to the learning process rather than focusing on the final code. Google Sinha pointed out: "If your assignment is designed so that students can easily look up the answer, then it may not be the best assignment design. Learning is far more than that"[6]. This view reminds course teachers to adjust new grading standards to guide students toward deep learning.

New evaluation methods in practice include:

Process-oriented evaluation: Focus on students' performance in abilities such as problem decomposition, debugging strategies, and resource utilization efficiency during the programming process, not just the final code.

Diversified ability evaluation: Incorporate abilities such as algorithmic thinking, code design, documentation writing, and collaborative communication into the evaluation scope.

AI-assisted evaluation: Use AI technology for real-time learning condition analysis, providing personalized feedback, while retaining teachers' humanistic evaluation and thinking guidance. These methodological changes collectively point to the multidimensional transformation of the evaluation system from concept to practice, as shown in Table 3:

Table 3: Multidimensional Changes in Python Course Evaluation System in the AI Era

Evaluation Dimension	Traditional Evaluation Method	AI-Era Evaluation Method
Evaluation Focus	Code correctness and efficiency	Computational thinking and innovative application
Evaluation Method	Programming exams and assignment checks	Diversified, personalized evaluation
Evaluation Subject	Single teacher evaluation	Joint teacher-student evaluation, including AI assistance
Evaluation Timeliness	Outcome-oriented evaluation	Process-oriented and developmental evaluation
Data Source	Final code and exam scores	Full programming process data collection and analysis

6. Conclusion and Outlook

Higher education in the AI era is facing severe challenges. In response to the problem of declining learning motivation among university students due to the notion that "knowledge mastered by AI need not be learned," teachers must comprehensively reform educational concepts, teaching methods, and evaluation systems. Taking Python courses as an example, the educational guidance path proposed in this paper focuses on transforming students from "code receivers" to "intelligent navigators" to help students rediscover the meaning and motivation of learning in the AI era.

AI is always an extension of human wisdom, not a substitute. In the current context of rapid technological updates, Python education must return to its essence-cultivating well-rounded individuals who can harness technology rather than being bound by it. By constructing a new ecosystem of human-machine harmonious Python education, we are expected to see that AI will not weaken learning motivation but instead become a powerful driving force for stimulating deep learning and innovative thinking. What Python language courses should cultivate are precisely those future talents who can harness intelligent tools to solve complex problems.

References

[1] Ministry of Education. National Report on the Construction of Artificial Intelligence Majors in Higher Education Institutions[R]. Beijing: Department of Higher Education, Ministry of Education, 2025.

[2] Tsinghua University Institute of Education. Tracking Study on the Impact of AI Learning Companions on University Students' Learning Outcomes[J]. Higher Education Research, 2024,45(3):56-62.

[3] MIT Teaching and Learning Laboratory. Experimental study on the use of ChatGPT in programming courses[R]. Cambridge, MA: Massachusetts Institute of Technology, 2024.

[4] Jiang Guohua. Learning Paradigm Transformation in the AI Era: From Knowledge Instillation to Ability Generation[J]. Education Research, 2024,45(2):34-41.

[5] Ryan R M, Deci E L. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being[J]. American Psychologist, 2000,55(1):68-78.

[6] Sinha S. How students are creatively using AI in education[EB/OL]. Google Education Blog, 2024.

[7] Fu Qiang. Cognitive Extension in the AI Era: From Knowledge Receiver to Intelligent Navigator[J]. China Higher Education, 2024,(5):23-27.

[8] Xi'an Jiaotong University. Innovative Practice of the "Three Assistance" Teaching Model under the Background of Digital Transformation[J]. Research in Higher Engineering Education, 2024,(3):45-51.

[9] Reform Framework for Higher Education Quality Improvement Task Force. Reform Framework for Higher Education Quality Improvement in the AI Era[J]. China Higher Education Research, 2024,(4):12-19.