ISSN 2522-6398 Vol. 9, Issue 9: 33-38, DOI: 10.25236/FER.2026.090905

Generative AI-Driven Reconstruction of the Programming Curriculum System for New Engineering Majors

Lili Zhang[a], Wei Wei[b], Jing Li[c], Wentao Wu[d], Ge Wang[e]

College of Information Engineering, Beijing Institute of Petrochemical Technology, Beijing, China
[a]zhanglili@bipt.edu.cn, [b]weiwei@bipt.edu.cn, [c]bipt_lijing@bipt.edu.cn, [d]wuwentao@bipt.edu.cn, [e]wangge@bipt.edu.cn
**Corresponding author*

Abstract: *Against the background of the rapid rise of generative artificial intelligence (GAI) represented by DeepSeek, ChatGPT, ERNIE Bot, Tongyi Qianwen, Doubao and Kimi, the teaching of programming courses of new engineering majors is facing profound challenges: the role of programmers is undergoing a fundamental transformation, the traditional "programming-centered" teaching content can hardly meet the industrial demand of the GAI era, and engineering practice is disconnected from the real production process of enterprises. To address these problems, this paper investigates the influence mechanism of GAI on the teaching of programming courses and proposes systematic improvement measures. First, starting from the feasibility and inevitability of GAI undertaking basic programming tasks, the coupling mechanism between GAI and programming teaching is analyzed, and the potential possibility of programming role transformation and the inevitable trend of human-machine collaborative development of complex functions are revealed. Second, the concepts of teaching and learning are reconstructed, and a "broad-specialized-integrated" programming curriculum system integrating the "design-centered" concept is constructed. Third, combined with the CDIO engineering education concept, a GAI-driven "design-centered" engineering practice model is explored, and an engineering case base highlighting disciplinary intersection and facing real industry scenarios is built. The research provides a theoretical reference and a practical path for the construction of programming courses of new engineering majors in application-oriented universities in the GAI era.*

Keywords: *Generative artificial intelligence; New engineering; Programming courses; Curriculum system; Engineering practice; CDIO*

1. Introduction

Programming courses occupy a pivotal position in the curriculum system of engineering majors: they constitute the skeleton of the whole curriculum system, support the main structure of students' knowledge architecture, serve as the key bridge connecting students with research-and-development enterprises and employers, and lay the foundation of product design and system development. In recent years, generative artificial intelligence (GAI) represented by DeepSeek, ChatGPT, ERNIE Bot, Tongyi Qianwen, Doubao and Kimi has risen rapidly and achieved major breakthroughs in key fields such as natural language processing and code generation[1-3]. This development has not only triggered extensive discussion in the field of science and technology, but also set off waves in the education community, exerting an extremely profound influence on programming posts in research-and-development enterprises and on the traditional teaching content of programming-related courses in universities, and bringing unprecedented challenges[4-5].

As early as 2018, the Artificial Intelligence Innovation Action Plan for Higher Education Institutions issued by the Ministry of Education clearly pointed out that colleges and universities should aim at the frontiers of world science and technology, and continuously improve their capabilities of scientific and technological innovation, talent cultivation and international cooperation and exchange in the field of artificial intelligence, so as to provide strategic support for the development of the new generation of artificial intelligence in China[6]. This measure has strongly promoted the development of artificial intelligence technology in higher education, especially in engineering education, from the national level. However, the rapid development and sudden emergence of GAI still caught the teachers

ISSN 2522-6398 Vol. 9, Issue 9: 33-38, DOI: 10.25236/FER.2026.090905

of core programming courses of engineering majors somewhat unprepared[7-8]. On one hand, teachers are faced with the problem of how to update the teaching content, because the traditional programming teaching content may not fully adapt to the requirements of the GAI era; on the other hand, teaching methods also need to be innovated to guide students to use GAI correctly to assist programming[9-10].

Taking the Internet of Things Engineering major of Beijing Institute of Petrochemical Technology—a Beijing first-class undergraduate major—as the carrier, and oriented to the development demands of “new industries” and “new occupations”, this paper systematically studies the influence mechanism of GAI on the teaching of programming courses of new engineering majors and the corresponding improvement measures. Using GAI to assist or even dominate programming will become an inevitable trend in the future development of the software and hardware research-and-development field; therefore, the urgent problem to be solved at present is how to clarify and construct the teaching concept, teaching content and engineering practice that adapt to this trend. The remainder of this paper is organized as follows: Section 2 analyzes the impact of GAI on the teaching of programming courses; Section 3 reveals the influence mechanism; Section 4 presents the reconstruction of the curriculum system integrating GAI; Section 5 elaborates the GAI-driven engineering practice model; Section 6 introduces the research methods, implementation path and expected outcomes; Section 7 concludes the paper.

2. The Impact of Generative Artificial Intelligence on the Teaching of Programming Courses

(1) The Fundamental Transformation of the Programmer’s Role

With the breakthrough of large language models in code generation, basic programming tasks—such as syntax implementation, function writing and simple debugging—can already be completed by GAI with high quality[3]. The role of programmers is therefore undergoing a fundamental transformation: from “code writers” who implement every line by hand to “designers and reviewers” who define requirements, design architectures, organize human-machine collaboration and review machine-generated code. In the research-and-development practice of enterprises, engineers who skillfully use GAI tools can complete the development work of several traditional programmers, and the mode of human-machine collaborative development of complex functions is becoming the mainstream[10]. This transformation directly impacts the objective orientation of programming courses: if the courses still take syntax memorization and manual coding proficiency as the core, the cultivated students will inevitably face the risk of being replaced by machines upon graduation[1].

(2) The Lag of the Traditional “Programming-Centered” Teaching Content

Traditional programming courses are designed around the logic of “programming-centered”: the teaching content focuses on language syntax, data structures and algorithm implementation, and the assessment emphasizes whether students can independently write correct code[4]. However, when GAI can generate syntactically correct code within seconds, the value of such teaching content is greatly weakened. On one hand, students can easily obtain assignment answers with the help of GAI, which makes traditional homework and examinations lose their discriminating power and brings new challenges to academic integrity[2]; on the other hand, the abilities really needed by the industry in the GAI era—requirement analysis, product design, prompt engineering, code review and system integration—are almost absent from the current curriculum system, resulting in a structural mismatch between course objectives and industrial demand[7].

(3) The Disconnection between Engineering Practice and Real Industrial Demand

In the GAI era, enterprises’ requirements for programming personnel have shifted from “being able to code” to “being able to deliver products with the collaboration of AI”. Nevertheless, the practical links of most programming courses still stay at the level of verification experiments and toy projects, which are far from the real production environment and research-and-development process of enterprises[9]. Students lack the training of the whole process of “requirement analysis–product design–human-machine collaborative development–testing and deployment”, and their abilities of engineering design, system integration, function development, installation and debugging, and operation and maintenance can hardly meet the actual needs of emerging fields such as new-generation information technology, autonomous driving and artificial intelligence[5]. How to build a practical training mode that highly restores the real production environment of enterprises with the help of GAI has become a key problem that engineering practice teaching must solve[8].

ISSN 2522-6398 Vol. 9, Issue 9: 33-38, DOI: 10.25236/FER.2026.090905

3. The Influence Mechanism of Generative Artificial Intelligence on Programming Courses

(1) The Feasibility and Inevitability of GAI Undertaking Basic Programming Tasks

Starting from the technical essence of GAI, the code generation capability of large language models originates from the probability modeling of massive code corpora, which makes them particularly good at basic programming tasks with clear patterns and sufficient samples[3]. From the perspective of feasibility, mainstream GAI tools can already pass the majority of introductory programming test cases and generate runnable engineering code fragments, which fully demonstrates their ability to undertake basic programming tasks. From the perspective of inevitability, driven by the economic law of efficiency first, the research-and-development field will inevitably tend to use GAI to complete standardized and repetitive coding work, while human programmers will inevitably move up to the links of requirement definition, architecture design and quality control. The superposition of the two determines that the introduction of GAI will lead to the potential possibility of the transformation of programming roles and the inevitable trend of human-machine collaborative development of complex functions[10].

(2) The Coupling Mechanism between GAI and Programming Teaching

The influence of GAI on programming teaching is not a one-way impact, but a two-way coupling. On the teaching side, GAI acts as a key teaching tool and resource that is integrated into the whole teaching process: teachers can use GAI to generate teaching cases, design hierarchical exercises and implement personalized tutoring, thereby reconstructing the teaching organization form[8]. On the learning side, students actively use GAI to acquire knowledge and solve problems, and the learning mode changes from “memorizing syntax” to “driving tools and reviewing results”. The two sides are coupled through the link of human-machine dialogue: the quality of students’ requirement description and prompt design determines the quality of the machine-generated results, and the review and correction of the generated results in turn promote the deepening of students’ understanding of programming principles[4]. This coupling mechanism reveals that the core of programming teaching in the GAI era is no longer the transfer of coding skills, but the cultivation of the ability to solve practical problems with the collaboration of GAI.

(3) The Influence Path on Teaching Objectives, Content and Evaluation

Through the above coupling mechanism, the influence of GAI is transmitted to three key links of the curriculum. First, at the level of teaching objectives, the course objective changes from imparting programming knowledge to cultivating students’ ability to use GAI to solve program development and practical problems. Second, at the level of teaching content, the content system must be deeply integrated with GAI, and new modules such as requirement analysis, product design and human-machine collaboration must be added on the basis of basic programming teaching, with the content and form optimized through application cases. Third, at the level of teaching evaluation, since machine-generated code is difficult to distinguish from students’ own work, the evaluation must shift from result evaluation to process evaluation, and measures such as on-site defense, development process recording and code review must be introduced to ensure the authenticity of learning outcomes and to prevent GAI from degenerating into a “ghostwriter”[2][10].

4. Reconstructing the Curriculum System Integrating Generative Artificial Intelligence

(1) Reconstructing the Concept of Teaching and Learning

The premise of curriculum system reconstruction is the reconstruction of the concept of teaching and learning. In teaching, GAI should be regarded as a key teaching tool and resource and be integrated into the whole teaching process in an all-round way, rather than being simply banned or passively accepted; in learning, students should be guided to take the initiative to acquire knowledge and solve problems with the help of GAI, and to establish a correct view of tool use—GAI is a “private tutor” that assists learning, not a “ghostwriter” that replaces thinking[10]. Teachers need to change from knowledge transmitters to designers of learning activities and guides of human-machine collaboration, while students need to change from passive recipients to active constructors, so as to jointly build a new teaching ecology of programming courses in the GAI era[1][7].

(2) Reconstructing the Teaching Content: From “Programming-Centered” to “Design-Centered”

ISSN 2522-6398 Vol. 9, Issue 9: 33-38, DOI: 10.25236/FER.2026.090905

In terms of teaching content, the concept of "design-centered" should be integrated into the traditional "programming-centered" curriculum system: on the basis of retaining the teaching of basic programming knowledge, the content of requirement analysis, product design and human-machine collaboration should be added. Specifically, in the introductory stage, while teaching language basics, students are trained to decompose real problems and describe requirements accurately; in the intermediate stage, teaching is organized around the complete life cycle of product design, and students learn to use GAI to complete scheme design, code generation and iterative optimization; in the advanced stage, complex engineering projects are introduced to train students' ability to organize human-machine collaborative development. With the assistance of GAI, the core competence elements of the whole process of program design facing the real post requirements of industrial enterprises are constructed, so that the teaching content can keep pace with the iteration of industrial demand[5][9].

(3) The "Broad-Specialized-Integrated" Curriculum Architecture

On the basis of the above reconstruction, a "broad-specialized-integrated" programming curriculum system is built. "Broad" refers to a broad foundation: through general modules such as computational thinking and GAI literacy, students of engineering majors master the basic concepts and application norms of GAI. "Specialized" refers to professional depth: aiming at the characteristics of majors such as Internet of Things Engineering, professional programming courses are deeply reconstructed, and the ability chain of "programming + AI + industry" is formed. "Integrated" refers to integration and penetration: through interdisciplinary projects, GAI is deeply integrated with professional scenarios, and a number of practical links highlighting disciplinary intersection and artificial intelligence are set up. The three levels are progressive and mutually supportive, which not only ensures a solid programming foundation, but also highlights the application-oriented school-running characteristics, and finally forms a new paradigm for the cultivation of programming skilled talents with GAI thinking in new engineering majors[6].

5. The Generative AI-Driven Engineering Practice Model

(1) The "Design-Centered" Practice Architecture Based on CDIO

Combined with the CDIO (Conceive-Design-Implement-Operate) engineering education concept, a GAI-driven "design-centered" engineering practice architecture is constructed. In the Conceive stage, students use GAI to conduct demand research and feasibility analysis; in the Design stage, students complete product scheme design with the assistance of GAI and review the design documents generated by the machine; in the Implement stage, human-machine collaborative development is carried out, in which GAI undertakes basic coding while students focus on architecture control, key algorithm design and code review; in the Operate stage, students complete system deployment, testing, operation and maintenance. GAI is deeply embedded in each stage as a capability amplifier, while students always occupy the dominant position of design decision-making and quality control, thus realizing the "design-centered" practice concept[9].

(2) The Engineering Case Base Facing Real Industry Scenarios

Around current affairs hotspots, a series of engineering practice cases highlighting disciplinary intersection and artificial intelligence are constructed, and an engineering case base facing real industry scenarios is formed. The case base highly restores the actual production environment and research-and-development process of enterprises: each case provides real requirement documents, constrained development conditions and acceptance standards, and students are required to complete the whole process of program design and development with GAI tools. The core ideas and key steps of using GAI to complete program design and development in different practical application scenarios are analyzed and summarized, and the typical experience is precipitated into teaching resources. Through the rolling update mechanism of the case base, the practical content can be dynamically iterated with the development of GAI technology and industrial demand, so as to ensure the advanced nature and authenticity of practical teaching[5].

(3) The Human-Machine Collaborative Development Workflow and Ability Evaluation

To support the operation of the practice model, a standardized human-machine collaborative development workflow is formulated: requirement description → prompt design → code generation → manual review → testing and verification → deployment and delivery, and the responsibility boundary between humans and machines at each node is clarified. Correspondingly, a multi-dimensional ability evaluation system is established, in which the evaluation objects are no longer limited to the code itself,

ISSN 2522-6398 Vol. 9, Issue 9: 33-38, DOI: 10.25236/FER.2026.090905

but cover the quality of requirement analysis, the rationality of design schemes, the effectiveness of prompt engineering, the rigor of code review and the completeness of system delivery. Through process recording, on-site defense and the participation of enterprise mentors, students' real ability in the human-machine collaborative environment is comprehensively investigated, so that the engineering practice projects based on GAI can effectively improve students' mastery of relevant knowledge and new technologies[2][8].

6. Research Methods, Implementation Path and Expected Outcomes

(1) Research Methods and Technical Route

Guided by the construction of new engineering and taking the cultivation of artificial intelligence technology application ability as the core, the research focuses on the influence mechanism of GAI on the teaching of programming courses and adopts a combination of four methods. First, the literature analysis method is used to collect, classify and compare domestic and foreign books and papers on the themes of "generative AI" and "programming courses", so as to clarify the existing problems and lay a theoretical research foundation. Second, the survey research method is used to design questionnaires and conduct visits and seminars, focusing on the exploration, practice and progress of other universities in applying GAI to programming course teaching, as well as the requirements of relevant key enterprises for programmers to apply GAI in program development and project implementation, and the feedback data and information are sorted and analyzed. Third, the case analysis method is used to learn from the application experience of GAI in education and curriculum construction by participating in relevant conferences and training. Fourth, the action research method is used to put the theoretical research into teaching practice, analyze and summarize the problems and experience in the practice process, and continuously optimize the construction scheme of programming courses.

(2) The Phased Implementation Path

The research is planned to be completed in two years in four stages. In the first stage (from January 2026 to June 2026), the application status of GAI in programming courses of relevant colleges and universities is investigated, the requirements of enterprises are understood through questionnaires and field visits, and the research scheme is continuously improved. In the second stage (from July 2026 to December 2026), the influence mechanism of GAI on programming courses is analyzed, the teaching concept and teaching content are reconstructed, engineering practice cases integrating GAI are developed, and the teaching practice of relevant courses is carried out. In the third stage (from January 2027 to June 2027), problems are discovered and analyzed through teaching practice, continuous improvement schemes are formulated, and the theoretical research of the project is optimized. In the fourth stage (from July 2027 to December 2027), the research contents and achievements are comprehensively summarized and the final research report is written. Through the closed-loop implementation of "research–practice–improvement–summary", the theoretical connotation and application value of the research are continuously perfected.

(3) Expected Outcomes

The research is expected to produce three educational teaching research papers and one research report. At the theoretical level, the influence mechanism of GAI on the teaching of programming courses is clarified, and a curriculum system integrating the concept of GAI is constructed, forming a new paradigm for the cultivation of programming skilled talents with GAI thinking in new engineering majors. At the practical level, based on the research results, programming courses are built and practiced to help students master the application of GAI in programming, change and cultivate students' brand-new programming thinking and ability, and use artificial intelligence to carry out practice and innovation. At the radiation level, the research results will provide theoretical reference and practical paths for the integration of new theories such as GAI into the construction of programming courses of new engineering majors in application-oriented universities, improve the matching degree between students' programming thinking and ability and the employment demand of industrial enterprises, and help the cultivation of innovative, applied and compound talents.

7. Conclusion

The rapid development of generative artificial intelligence is reshaping the programming posts of research-and-development enterprises and the teaching form of programming courses in universities.

ISSN 2522-6398 Vol. 9, Issue 9: 33-38, DOI: 10.25236/FER.2026.090905

Taking the teaching reform of programming courses of new engineering majors as the object, this paper analyzes the coupling mechanism and influence path of GAI on programming teaching from the feasibility and inevitability of GAI undertaking basic programming tasks, proposes the reconstruction scheme of the “broad-specialized-integrated” curriculum system integrating the “design-centered” concept, and designs the GAI-driven engineering practice model based on the CDIO concept. The research shows that the key to the reform of programming courses in the GAI era lies in the transformation from knowledge imparting to ability cultivation, from “programming-centered” to “design-centered”, and from passive response to active integration. Only by clarifying the influence mechanism and systematically reconstructing the teaching concept, teaching content and engineering practice can the programming courses of new engineering majors cultivate engineering application-oriented technical talents who meet the needs of the times. The influence of GAI on education is still evolving rapidly, and issues such as the dynamic iteration of curriculum content and the authenticity guarantee of learning evaluation deserve continuous in-depth investigation by teaching administrators and front-line teachers alike.

Acknowledgement

This work is supported in part by General Project of Beijing City Higher Education Society in 2025 (Project Number: 125) and Educational and teaching reform and research project funding of Beijing University of Petroleum and Chemical Technology (Research on Innovation and Practice of the Postgraduate Training Model of “Two-wheel Drive·Six-dimensional Connection”: 24033981001/027).

References

[1] Bi, W. X. Challenges of Generative Artificial Intelligence to the Education Industry and the Responses: Taking ChatGPT as the Analysis Object. Jiangsu Higher Education, 2023, (08): 13-22.
[2] Wei, S. P., Fan, X. J., Wang, X. X., et al. The Potential and Risks of Applying ChatGPT in Higher Education: Experience and Enlightenment from American Universities. Modern Distance Education, 2024, (03): 18-27.
[3] Tao, W., Shen, Y. From ChatGPT to Sora: Four-Ability Education and Paradigm Innovation Oriented to AIGC. Modern Educational Technology, 2024, 34(04): 16-27.
[4] Zheng, Y. H., Zhou, D. H., Zhang, Y. H., et al. ChatGPT from the Perspective of Computational Pedagogy: Connotation, Themes, Reflection and Challenges. Journal of East China Normal University (Educational Sciences), 2023, 41(07): 91-102.
[5] Qi, Y. Y., Wang, L. M. Generative Artificial Intelligence Applied in Open Education: Opportunities, Challenges and Application Scenarios. Adult Education, 2024, 44(06): 56-61.
[6] Ministry of Education. Notice on Issuing the Artificial Intelligence Innovation Action Plan for Higher Education Institutions. 2018-04-02.
[7] Lu, Y., Yu, J. L., Chen, P. H., et al. Research and Prospect of the Educational Application of Multimodal Large Models. E-education Research, 2023, 44(06): 38-44.
[8] Huang, R. H. Integrating Large Artificial Intelligence Models into Education: Conceptual Transformation, Form Reshaping and Key Measures. Frontiers, 2024, (14): 23-30.
[9] Wu, D., Li, H., Chen, X. An Analysis of the Influence of Artificial Intelligence General Large Models on Educational Application. Open Education Research, 2023, 29(02): 19-25+45.
[10] Li, Q. Y., Gengyang, L. A., Peng, W. J., et al. “Private Tutor” or “Ghostwriter”: Exploration of Computer Practical Teaching Based on Large Models. Experimental Technology and Management, 2024, 41(05): 1-8.