RESEARCH ARTICLE

Exploration of the Reform of Comprehensive Practice Course in Advanced Programming Driven by Large Models

Nan Guo[1,*], **Bing Han**[1] **and Junfei Qiao**[1]

[1] School of Information Science and Technology, Beijing University of Technology, Beijing 100124, China

Abstract

The rapid advancement of AI and large language models has placed newer and higher demands on comprehensive programming practice courses in higher education. Significant shifts in students' learning styles, knowledge acquisition channels, and innovative capabilities have rendered traditional curriculum content and teaching methods inadequate for meeting the talent development requirements of the new era. This paper thoroughly examines the profound impact of the widespread adoption of AI technologies on the curriculum system and proposes a systematic reform framework centered on "AI empowerment, competency orientation, and student-centered approaches." By integrating AI tools into the course, it aims to enhance the efficiency and innovation of students' programming practice, strengthen their ability to use intelligent tools rationally and critically, emphasize interdisciplinary integration and engineering ethics, and ultimately foster students' capacity to solve complex engineering problems systematically.

Submitted: 31 October 2025
Accepted: 13 November 2025
Published: 25 December 2025

Vol. 1, **No.** 3, 2025.
10.62762/FEIR.2025.722615

***Corresponding author:**
Nan Guo
guonan03@bjut.edu.cn

Keywords: artificial intelligence, large language models, curriculum reform, competency-oriented.

1 Introduction

Breakthroughs in artificial intelligence and large models are profoundly reshaping the landscape of higher education and its learning ecology [1]. In response to this trend, the "Advanced Programming Comprehensive Practice" course—which focuses on cultivating practical abilities—stands at the forefront of educational transformation, facing significant influences from both pedagogical philosophies and evolving talent demands. Students now exhibit new characteristics in their learning approaches, channels of knowledge acquisition, and innovative capabilities, making traditional teaching models increasingly inadequate for meeting the practical requirements of talent cultivation in the new era. Specifically, the widespread adoption of AI tools has significantly enhanced students' efficiency in basic learning tasks such as syntax queries and code generation, fundamentally altering their pathways to acquiring knowledge [2].

The contemporary technological landscape has catalyzed a fundamental shift in industry expectations.

Citation
Guo, N., Han, B., & Qiao, J. (2025). Exploration of the Reform of Comprehensive Practice Course in Advanced Programming Driven by Large Models. *Frontiers in Educational Innovation and Research*, 1(3), 81–91.

Employers no longer prioritize mere "coding skills" but instead seek candidates with sophisticated "complex engineering problem-solving abilities." This paradigm shift emphasizes the critical need for students to develop multifaceted competencies spanning algorithm design, system architecture, optimization methodologies, and interdisciplinary integration. The ability to synthesize knowledge across domains and apply computational thinking to novel contexts has become paramount in industry settings where technological boundaries are increasingly fluid.

In parallel with these changing industry demands, the educational environment faces unprecedented disruption from generative AI technologies. Large language models capable of producing functional code, debugging existing programs, and explaining complex computational concepts have democratized access to programming knowledge while simultaneously challenging traditional pedagogical approaches. These developments have rendered conventional teaching methodologies—characterized by syntax-focused instruction and predefined problem sets—increasingly obsolete in preparing students for real-world challenges where contextual understanding and creative problem formulation are essential.

The current social context, marked by the rapid evolution of large model technologies and the ubiquitous integration of AI tools across educational and professional domains, presents the "Advanced Programming Comprehensive Practice" course with a constellation of interconnected challenges, primarily manifested in the disconnect between teaching content and cutting-edge practice, the diminished effectiveness of traditional teaching methods, and the inadequacy of existing evaluation systems in accurately assessing real abilities. This series of issues indicates that existing teaching models are no longer sufficient to meet the actual needs of cultivating high-level engineering talent in the new era.

This constellation of challenges underscores that existing teaching models are fundamentally insufficient to meet the actual needs of cultivating high-level engineering talent in the new era. The convergence of technological iteration, shifting industry demands, and the limitations of traditional pedagogical approaches collectively constitutes the core driving force behind curriculum reform.

The transformation required extends beyond incremental adjustments to encompass a comprehensive reimagining of programming education. This reimagining must acknowledge AI not merely as a disruptive force but as a potential catalyst for pedagogical innovation. By strategically integrating AI tools into educational frameworks while simultaneously developing students' critical faculties for evaluating and extending AI-generated solutions, educators can create learning environments that prepare students for the complex socio-technical systems they will encounter professionally.

Looking ahead, building a systematic teaching reform path centered on "AI empowerment" and "competency orientation" has become an essential requirement for achieving innovation and high-quality development in advanced programming education. This reform path must balance technological fluency with conceptual depth, emphasize authentic problem-solving in complex contexts, and develop students' capacities for ethical reasoning and critical evaluation of technological solutions. Only through such comprehensive reform can programming education fulfill its essential role in preparing the next generation of computational thinkers for the challenges and opportunities of an AI-augmented future.

2 Analysis of Problems in Traditional Teaching Models

Accurately identifying the gap between the "Advanced Programming Comprehensive Practice" course and the talent cultivation requirements of the artificial intelligence era requires a systematic examination of its current teaching model. As shown in Figure 1, the course system demonstrates substantial disconnects from cutting-edge technological developments and current industry demands across three key dimensions. Specifically, the teaching content lags behind technological practices, the teaching methods remain inadequate for human-machine collaboration paradigms, and the evaluation system fails to effectively foster higher-order competencies [3].

2.1 Misalignment Between Teaching Content and Contemporary Demands

The knowledge system of the "Advanced Programming Comprehensive Practice" course is updated slowly and lacks agility in responding to the rapidly evolving requirements for developer competencies in the artificial intelligence era. While the field of programming is undergoing significant transformation—driven by the integration of AI, big data, and interdisciplinary applications—the course

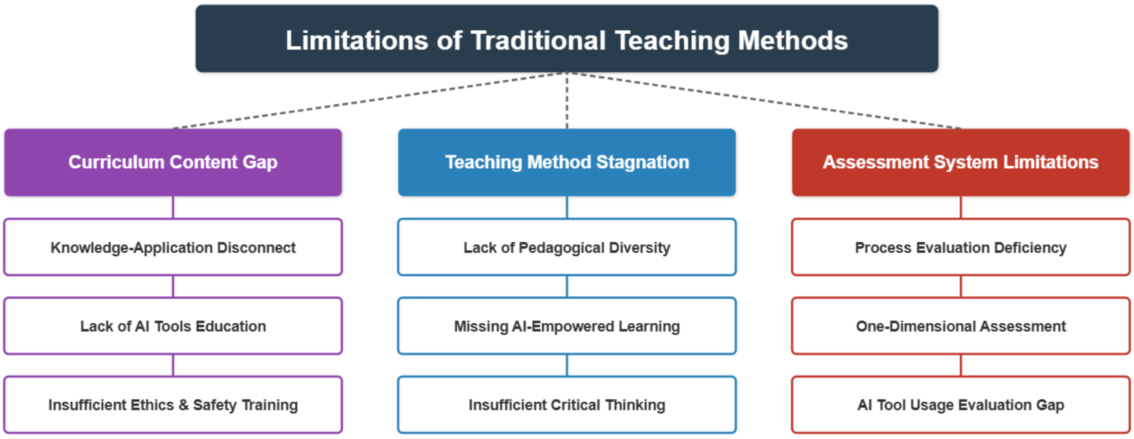

Figure 1. Problems in traditional teaching of the course.

content remains excessively focused on syntactic details of specific programming languages and isolated algorithms. This narrow emphasis means students are primarily trained in basic coding skills, but rarely exposed to real-world application scenarios from specialized domains such as bioinformatics, financial computing, and intelligent systems. As a result, they struggle to apply their knowledge to solve complex, multidisciplinary problems and often find themselves "knowing code but not understanding applications," with severely limited capability for knowledge transfer [4].

Moreover, the curriculum has yet to systematically incorporate instructions on the usage, efficacy boundaries, and ethical considerations of intelligent programming assistants, such as large language models [5]. In today's context, where AI-assisted programming is becoming mainstream, this omission creates a critical gap. Students are not provided with structured guidance on how to leverage these tools effectively and responsibly, nor are they taught to discern the strengths and limitations of AI-generated solutions. Consequently, their understanding of AI-assisted programming remains superficial, which hinders the development of efficient problem-solving skills through human-AI collaboration.

Additionally, the course's overemphasis on technical implementation and outcome orientation leads to insufficient attention to critical issues such as code security, data privacy, and algorithmic fairness. In the current landscape, where software vulnerabilities and unethical algorithms can have far-reaching societal consequences, neglecting these aspects leaves students with notable deficiencies in engineering ethics awareness and safety standards comprehension. This not only limits their professional readiness but also fails to meet the expectations of modern industry, which increasingly values responsible and ethical programming practices.

2.2 Lagging Innovation in Teaching Methods

The current teaching methodology of the course predominantly adheres to traditional paradigms, with little adaptation to the new technological ecosystem represented by artificial intelligence. This rigidity results in an increasing disconnection from contemporary students' cognitive patterns and learning habits. The instructional approach remains primarily "teacher-led and student-imitated," where students passively receive information rather than actively engage with the material. There is inadequate implementation of student-centered methods such as project-based and inquiry-based learning, which are essential for stimulating students' initiative, curiosity, and innovative potential. Without opportunities to explore, experiment, and collaborate, students are deprived of the chance to develop adaptive and self-directed learning abilities—attributes that are crucial for success in the fast-changing field of programming.

Furthermore, the course has not established a systematic AI-enhanced learning framework. It fails to guide students in utilizing intelligent tools like large language models as powerful assistants for efficient learning and problem-solving. In the absence of structured instruction, students either avoid using these tools altogether, missing out on valuable learning opportunities, or use them indiscriminately, without understanding their capabilities, limitations, or potential biases. This lack of guidance prevents the deep integration of human-AI collaboration into the learning process, which is increasingly necessary as AI technologies become embedded in professional programming environments.

Additionally, the teaching process places insufficient emphasis on developing critical thinking skills. The curriculum often prioritizes role memorization of syntax and algorithms over cultivating analytical reasoning, problem decomposition, and reflective thinking. As a result, students generally lack the ability to evaluate, optimize, and reflect on AI-generated content, leading to overreliance on tool outputs and diminished capacity for independent judgment and creative problem-solving [6]. When faced with novel or ambiguous problems that require innovative approaches, students may struggle to move beyond the solutions provided by AI tools, highlighting a gap in their metacognitive and higher-order thinking skills.

2.3 Evaluation System in Urgent Need of Improvement

The existing assessment framework proves inadequate in both assessing and fostering students' capabilities in solving complex engineering problems. Current evaluation practices tend to overemphasize final examinations or completed programming products, placing the main focus on the end results rather than the processes involved in requirements analysis, solution design, and iterative debugging. This outcome-focused assessment approach fails to capture the developmental journey of students' problem-solving capabilities and provides limited insights into their reasoning processes. With the widespread adoption of generative AI in programming education, the negative impact of such assessment bias has become even more pronounced. Recent studies show that 77% of students frequently use large language models (LLMs) to assist with programming tasks, yet the current assessment system does not sufficiently address students' processes of verifying, debugging, and integrating AI-generated code. As a result, some students become overly reliant on tool outputs while neglecting essential cognitive training [7]. Moreover, assessments that focus solely on outcomes fail to reflect students' application of critical thinking in human-AI collaboration, such as logical validation and ethical considerations regarding AI-generated solutions—capabilities that are now regarded as core requirements for programming professionals in the industry [8]. As a result, students are encouraged to prioritize "getting the right answer" instead of developing robust methodological approaches, which impedes the cultivation of resilient engineering habits and systematic problem-solving skills through repeated experimentation. This disconnect between assessment and practice not only undermines the quality of talent development in programming education but also makes it difficult for students to adapt to authentic "human-AI collaboration" scenarios in the AI era.

Moreover, the grading criteria are oversimplified, prioritizing surface-level metrics like code "correctness" and "efficiency" while neglecting essential engineering competencies such as modular design rationality, architectural integrity, code maintainability, and result interpretability. Particularly with the widespread integration of AI tools in learning and development, the current evaluation framework fails to adequately address this transformation. It provides insufficient guidance for students on properly utilizing AI to enhance efficiency and lacks a systematic approach to evaluating their ability to discern, verify, and integrate AI-generated outputs. Consequently, a significant gap exists between assessment practices and genuine engineering innovation.

In summary, the misalignment between teaching content and industry needs, the lagging innovation in teaching methods, and the inadequacy of the evaluation system collectively undermine the course's effectiveness in preparing students for successful careers in the rapidly evolving field of software development.

3 Exploring Reform Measures for Programming Courses in the AI Era

In response to the evolving educational landscape in the AI era, this study proposes a transformative instructional framework integrating "AI empowerment, student-centered approaches, and competency-based education." As illustrated in Figure 2, our reform strategy addresses four critical dimensions. These include restructuring teaching content, innovating teaching methods, reforming assessment systems, and building practice ecosystems. Together, these dimensions form a comprehensive approach to curriculum reform that bridges theoretical knowledge with practical application in the intelligent era. To systematically evaluate the effectiveness of these reform measures, the study also outlines an experimental design and data processing methods, providing a methodological foundation for subsequent

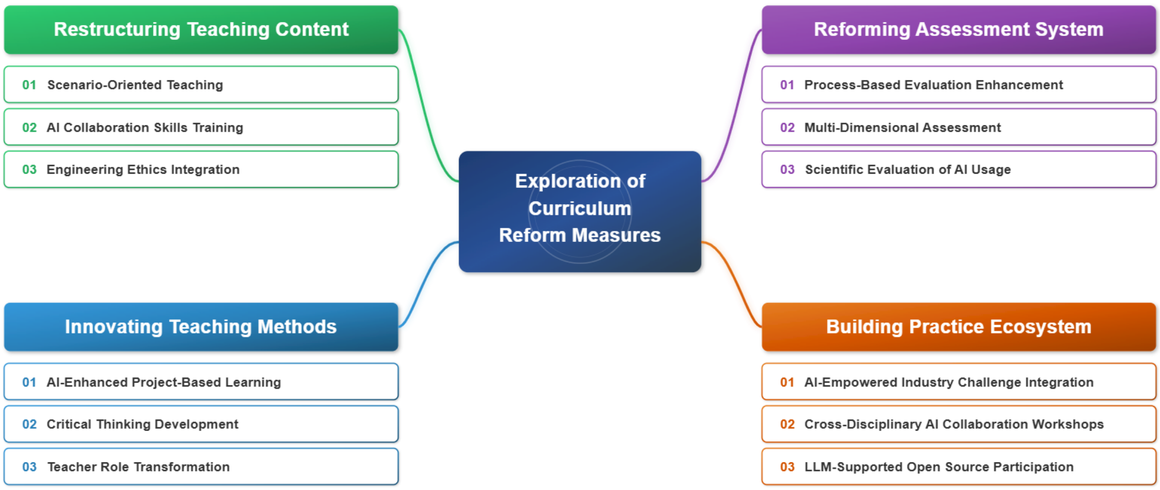

Figure 2. Exploration of curriculum reform measures.

empirical analysis.

3.1 Building an "AI-Augmented" Course Knowledge System

To address the limitations of the traditional syntax-focused curriculum, we advocate for a knowledge system that emphasizes "scenario empowerment." This means moving beyond isolated language and algorithm instruction, and instead designing project-based learning units rooted in cutting-edge application scenarios—including intelligent systems, scientific computing, and interdisciplinary domains. By embedding the study of algorithms and data structures within authentic problem-solving contexts, students are encouraged to apply foundational concepts in diverse and dynamic environments, thereby enhancing their ability to transfer knowledge and innovate across disciplinary boundaries.

A key innovation is the introduction of an "AI Collaborator" module as a central pillar of the curriculum. This module will provide systematic training in the effective use of large language models and similar AI tools for a variety of tasks, such as prompt engineering, code generation, debugging optimization, and documentation retrieval. Students will learn not only how to harness these tools to improve their productivity and problem-solving efficiency, but also how to critically assess their outputs, understand their limitations, and recognize potential ethical risks. The curriculum will clarify the applicable scope of AI tools, establish quality verification methods, and highlight the importance of responsible usage, positioning AI as a rational and controllable assistant [9].

Furthermore, engineering ethics and safety education will be deeply embedded into the instructional design. Issues such as code security, data privacy, algorithmic fairness, and traceability will be treated as mandatory constraints and evaluation criteria in all project practices. Through contextualized case studies, group discussions, and formative feedback mechanisms, students will be guided to develop risk awareness, ethical judgment, and social responsibility. This comprehensive approach ensures that students not only master technical skills but also understand the broader impact of their work, forming a future-oriented, industry-aligned "AI-Augmented" curriculum framework that prepares them for the ethical and practical demands of intelligent development environments.

3.2 Implementing an Inquiry-Based "Human-AI Collaborative" Teaching Model

To foster deeper learning and practical competence, the teaching model will shift toward inquiry-based, human-AI collaborative approaches. Multi-level, open-ended project tasks will be designed to encourage students to actively engage with AI tools for foundational development, information retrieval, and preliminary solution comparison. By automating routine tasks, students can allocate more cognitive resources to complex activities such as system architecture design, integration testing, and performance optimization. This division of labor enables effective human-AI synergy, promoting both the depth of learning and the quality of practical outcomes.

Critical thinking will be systematically integrated throughout the instructional process. During project evaluations and code reviews, students will be required to submit validation reports and structural analyses of AI-generated code. This practice not only enhances their technical assessment skills, but also cultivates the ability to make informed decisions and maintain rigorous standards when working with AI outputs [10]. By regularly reflecting on the strengths and weaknesses of automated solutions, students become more discerning users of technology, reducing the risk of overreliance and fostering independent problem-solving capabilities.

The reform also redefines the roles of teachers and students. Instructors will transition from being mere transmitters of knowledge to designers of engaging learning environments, facilitators of inquiry, and mediators of human-AI interaction. Their focus will shift toward shaping students' cognitive frameworks and fostering higher-order competencies, including critical and innovative thinking. Students, in turn, will be empowered to take greater ownership of their learning journeys, actively exploring new technologies and collaboratively solving complex problems [11]. This dynamic, student-centered environment nurtures adaptability and creativity—qualities essential for success in the AI-driven era.

3.3 Establishing a Comprehensive "Whole-Process, Multi-Dimensional" Assessment Mechanism

To align assessment practices with the goals of competency-based education, the course evaluation system will be fundamentally reformed. Greater emphasis will be placed on formative assessment and continuous competency development, with a corresponding reduction in the weight of final summative examinations. A learning portfolio will be implemented to systematically document students' progress across key phases—including requirements analysis, solution design, AI tool application, iterative debugging, and final implementation. This portfolio will serve as a record of students' evolving problem-solving strategies, tool utilization techniques, and reflection on learning experiences.

The evaluation framework moves beyond the conventional PBL emphasis on functional implementation and code correctness by introducing a more contemporary "human-AI collaboration" dimension. While retaining professional metrics such as code maintainability and system architecture rationality, the new framework prioritizes assessing students' effectiveness in leveraging AI tools to solve problems, their critical discernment of AI-generated outputs, and their adherence to ethical norms in human-AI collaborative processes. This approach aims to construct a comprehensive profile of programming competence reflective of the intelligent era. Furthermore, to align with the collaborative dynamics of modern human-AI teaming environments, the evaluation system will incorporate a peer review mechanism focused on the collaborative process.

Importantly, the assessment criteria will explicitly encourage the appropriate and effective use of AI tools. Students will be evaluated not only on whether they use intelligent assistants, but on how thoughtfully and critically they integrate these tools into their workflow. Specific focus will be placed on their ability to verify, refine, and synthesize AI-generated outputs, supported by critical reflection on their application. This shift in emphasis—from simple tool usage to the quality and responsibility of usage—aims to guide students toward becoming innovative practitioners capable of leveraging intelligent tools with discernment and ethical awareness [12].

3.4 Building a "Industry-Education Integration and Open Co-creation" Practice Ecosystem

A major challenge identified in the analysis of existing course problems is the disconnect between students' acquired knowledge and actual engineering practice, particularly the lack of deep integration with authentic, real-world application scenarios. In the context of rapid advancements in AI and large language model technologies, this gap becomes even more pronounced. Traditional programming education, with its focus on abstract knowledge and isolated skills, struggles to keep pace with technological transformations, leaving students ill-prepared for the demands of intelligent development environments [13].

To address this issue, the reform initiative seeks to construct an open, industry-education integration ecosystem through several interconnected mechanisms. First, deep collaborative relationships will be established with industry-leading enterprises, enabling the regular introduction of real engineering challenges—incorporating AI technologies—as course project materials. These projects will span various complexity levels, allowing students to apply their knowledge in contemporary engineering contexts and gain firsthand experience with how large language models are reshaping software development processes and methodologies [14]. Second, cross-disciplinary collaborative workshops centered on AI technologies will be organized, inviting experts from fields such as bioinformatics, financial computing, and intelligent systems to engage with students. These workshops will facilitate exploration of specialized programming applications empowered by AI, helping students understand domain-specific requirements and challenges, and overcoming the dilemma of "knowing code but not understanding applications." By interacting with professionals from multiple disciplines, students gain valuable insights into the practical relevance of their studies and the broader impact of intelligent technologies [15]. Third, institutionalized mechanisms will be developed to encourage student participation in AI-related open-source projects. Students will be guided to contribute code, submit issue reports, or engage in documentation writing with the assistance of large language models. This hands-on experience allows them to participate in the complete lifecycle of software development in the intelligent era, within authentic collaborative environments. Through these activities, students not only consolidate their technical skills but also develop teamwork, communication, and project management abilities essential for modern engineering practice.

In the AI era, programming education must evolve from merely imparting specific languages and algorithms to cultivating students' comprehensive abilities to solve complex problems with the assistance of intelligent tools. By leveraging the industry-education integration ecosystem, students can verify and apply their knowledge in AI-driven development environments that closely approximate real-world conditions. They will master human-machine collaborative work models, thereby nurturing new engineering talents capable of maintaining sustained competitiveness amid the intelligent transformation. This reform not only addresses the current disconnection between coursework and industry practice, but also represents a forward-looking exploration of the future direction of programming education in the context of intelligent transformation.

3.5 Experimental Design and Data Processing Methods

To provide a rigorous framework for evaluating the effectiveness of the proposed curriculum reform, this study outlines an experimental design that incorporates both an experimental group and a control group. The intended participants are undergraduate students with basic programming skills. The sampling strategy involves randomly selecting students from two parallel classes of the same grade. One class is planned to serve as the experimental group, participating in the AI-empowered, student-centered curriculum; the other class will act as the control group, following traditional teaching methods. Prior to grouping, statistical tests will be conducted to ensure that there are no significant differences between the groups in terms of gender, academic performance, and programming experience. Students with low attendance rates will be excluded from the analysis to enhance data validity.

For the analysis of AI tool usage, the design includes systematic collection of usage logs through the learning platform. The data processing plan consists of several steps: first, removing anomalous or incomplete records caused by network issues or operational errors; second, merging duplicate actions to avoid repeated counting; third, anonymizing all logs to protect student privacy; and finally, retaining only those entries directly related to course activities for subsequent analysis. This data cleaning process is intended to ensure the reliability and accuracy of behavior and outcome evaluation.

Through this experimental design and data processing framework, the study aims to establish a solid methodological foundation for future empirical research on the integration of AI tools in advanced programming education.

4 Future Prospects and Implementation Planning

In the context of deep integration of artificial intelligence into education, this curriculum reform is designed to empower students to use AI tools rationally and effectively. The ultimate goal is not only to enhance students' programming efficiency and foster their innovation capability, but also to cultivate systematic thinking and independent judgment. Rather than encouraging simple reliance on or passive acceptance of AI technologies, the reform aims to guide students in becoming proactive, discerning, and adaptive users of intelligent tools. Moving forward, the course will undergo continuous refinement, always guided by the core principles of "competency orientation" and "intelligent collaboration." Improvements will be systematically implemented in teaching content, pedagogical methods, and evaluation mechanisms. The long-term objective is to equip students with the ability to leverage intelligent tools, exercise independent critical thinking, and engage in lifelong learning, ensuring their sustainable competitiveness in the rapidly evolving technological landscape.

4.1 Cultivating the Ability to Master and Critically Apply AI Tools

The course reform envisions the establishment of a student-centered, AI-assisted diversified learning ecosystem that fundamentally transforms the traditional educational paradigm. Within this framework, instructors will transition from the role of mere knowledge transmitters to facilitators, mentors, and collaborators. Their primary responsibility will be to design challenging project tasks that stimulate students' curiosity, analytical thinking, and creative problem-solving abilities. By fostering an environment of inquiry and exploration, teachers will guide students in making informed decisions and in effectively utilizing AI tools to address authentic, complex problems encountered in real-world engineering contexts. Furthermore, instructors will help students develop structured frameworks for critical judgment, enabling them to evaluate the reliability and validity of AI-generated solutions.

The student learning process will be organized around a closed-loop cycle of "problem analysis, AI assistance, result validation, and solution optimization." The core innovation lies in repositioning AI as a "cognitive partner" rather than a passive tool. In this framework, AI forms a dynamic collaborative relationship with students throughout the problem-solving process. Students begin by analyzing problems, then engage with their AI cognitive partner through carefully designed prompts and feedback loops. Unlike traditional tool usage, students develop metacognitive abilities to critically evaluate AI outputs, identify limitations, and continuously refine solutions. This human-machine cognitive partnership not only improves programming efficiency but cultivates higher-order thinking as students interact with AI. Over time, students develop essential habits of questioning, verifying, and optimizing—forming cognitive enhancement patterns crucial for engineers in intelligent environments.

To ensure the curriculum remains relevant and forward-looking, course content and project topics will be dynamically updated to incorporate the latest AI application scenarios from industry frontiers, as well as to address students' evolving learning needs. This agile approach guarantees that students are exposed to cutting-edge technologies and real-world challenges, enhancing their ability to adapt and innovate. In addition, special emphasis will be placed on engineering ethics, safety standards, and the boundaries of tool usage. Through dedicated modules, case studies, and critical discussions, students will gain a balanced understanding of both the powerful capabilities and inherent limitations of AI technologies. They will learn to recognize ethical dilemmas, respect privacy and security requirements, and appreciate the societal impact of their technical decisions, thus growing into responsible and ethical practitioners.

4.2 Implementation Planning

To realize the above goals and ensure the effectiveness of the curriculum reform, a phased and systematic implementation plan will be adopted. The initial phase will focus on the deep integration of AI tools within the existing course structure. This includes refining implementation processes and standards, developing targeted instructional materials, and establishing a competency-centered teaching framework that aligns with the new educational objectives. During this stage, formative assessment will be strengthened to monitor students' progress in key areas such as requirements analysis, tool selection, and solution validation. By providing timely feedback and support, instructors will help students develop systematic habits of rational AI utilization and encourage reflective learning practices.

As the course model matures and initial reform measures demonstrate positive outcomes, the second phase will extend successful practices for AI tool application to related courses across the curriculum. This expansion will promote interdisciplinary integration and innovation, encouraging faculty and students from different academic domains to collaboratively explore optimal AI utilization strategies through diverse project-based learning experiences. Cross-disciplinary workshops, joint research projects, and collaborative teaching initiatives will be organized to foster a culture of openness, sharing, and continuous improvement. Instructors will be provided with professional development opportunities to enhance their understanding of AI technologies and pedagogical best practices, further supporting the sustainable advancement of the reform.

The final phase will focus on consolidating reform outcomes by developing transferable teaching resources and establishing standardized evaluation criteria. Comprehensive teaching guides, project templates, and case libraries will be created to facilitate the dissemination of successful experiences and methodologies. Through online open courses, systematic case sharing, and academic exchanges, the principles of rational AI utilization and competency-based education will be promoted throughout the engineering education community. This will help cultivate a new generation of professionals who are capable of leveraging intelligent tools effectively, exercising independent critical thinking, and sustaining innovation capacity in diverse contexts.

Confronted with the transformative impact of artificial intelligence, the "Advanced Programming Comprehensive Practice" course reform is systematically advancing through content reconstruction, pedagogical innovation, and evaluation enhancement. This phased implementation enables students to not only accomplish fundamental tasks efficiently, but, more importantly, to develop capabilities for mastering intelligent tools and pursuing continuous innovation. Looking forward, this reform will continue steering engineering education toward competency-based development, emphasizing openness, resource sharing, and cross-disciplinary integration. By maintaining a dynamic and adaptive approach, the curriculum will remain responsive to technological advances and industry trends, ensuring that students are well-prepared to thrive in the intelligent era.

5 Conclusion

This paper systematically explores curriculum reform for comprehensive programming practice courses in higher education against the backdrop of rapidly developing AI and large language model technologies. Through analyzing the limitations of traditional teaching models across three dimensions—teaching content, teaching methods, and evaluation systems—we propose an educational reform framework centered on "AI empowerment, student-centered learning, and capability orientation."

The reform measures encompass four key aspects. The first involves constructing an "AI-Augmented" knowledge system that goes beyond traditional syntax and algorithm instruction, integrating real-world scenarios and interdisciplinary applications to deepen students' understanding and practical skills. The second implements an inquiry-based teaching model featuring human-machine collaboration, where students actively engage with AI tools in project-based learning environments, fostering both technical proficiency and critical thinking. The third establishes a comprehensive, multi-dimensional assessment mechanism spanning the entire learning process, which incorporates formative and summative evaluations, peer reviews, and portfolio-based assessments to holistically measure students' competency development. The fourth creates a practical ecosystem characterized by industry-education integration and open co-creation, enabling students to participate in authentic engineering projects, collaborate with industry experts, and experience the full cycle of software development in intelligent environments. These measures collectively form a cohesive whole aimed at cultivating students' ability to solve complex engineering problems in the intelligent era.

By incorporating AI tools to assist learning and practice, students can potentially not only improve programming efficiency but, more importantly, strengthen their rational mastery and critical judgment of intelligent tools. Scenario-based instructional design helps promote knowledge transfer and application across interdisciplinary contexts, while process-oriented evaluation mechanisms can effectively guide students to focus on developing engineering thinking and comprehensive competencies. Furthermore, the integration of engineering ethics and safety standards into the curriculum ensures that students develop a responsible attitude toward technology usage, understand the broader societal impact of their work, and are equipped to make ethical decisions in complex, real-world situations.

In the future, we plan to gradually implement these reform measures in actual teaching, continuously optimize the "AI-empowered" teaching model, strengthen deep collaboration with industry, and explore more effective talent cultivation pathways. This will involve iterative refinement of curriculum content, regular updates to project scenarios reflecting the latest technological advancements, and ongoing professional development for instructors to stay abreast of emerging AI trends. Meanwhile, we will closely monitor AI technology development trends, adjusting teaching strategies in a timely manner to ensure that course content and teaching methods can adapt to the new requirements for high-level engineering talent in the intelligent era. We anticipate that this flexible and adaptive approach will enable the curriculum to remain relevant and effective, preparing students to meet future challenges with confidence and creativity.

The theoretical exploration of this research indicates that in the AI era, programming education is no longer merely about teaching specific languages and algorithms, but about cultivating students' comprehensive ability to solve complex problems with the assistance of intelligent tools. Through systematic curriculum reform, we hope to provide valuable reference for cultivating innovative talents who can adapt to and lead future technological transformations.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported by the National Natural Science Foundation of China under Grant 62303027.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Peláez-Sánchez, I. C., Velarde-Camaqui, D., & Glasserman-Morales, L. D. (2024, June). The impact of large language models on higher education: exploring the connection between AI and Education 4.0. In *Frontiers in Education* (Vol. 9, p. 1392091). Frontiers Media SA. [Crossref]

[2] Vemula, S. (2024, October). Enriching Python Programming Education With Generative AI: Leveraging Large Language Models for Personalized Support and Interactive Learning. In *2024 IEEE Frontiers in Education Conference (FIE)* (pp. 1-8). IEEE. [Crossref]

[3] Laato, S., Morschheuser, B., Hamari, J., & Björne, J. (2023, July). AI-assisted learning with ChatGPT and large language models: Implications for higher education. In *2023 IEEE International Conference on Advanced Learning Technologies (ICALT)* (pp. 226-230). IEEE. [Crossref]

[4] Scholl, A., & Kiesler, N. (2024, October). How Novice Programmers Use and Experience ChatGPT when Solving Programming Exercises in an Introductory Course. In *2024 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE. [Crossref]

[5] Kizilcec, R. F., Huber, E., Papanastasiou, E. C., Cram, A., Makridis, C. A., Smolansky, A., ... & Raduescu, C. (2024). Perceived impact of generative AI on assessments: Comparing educator and student perspectives in Australia, Cyprus, and the United States. *Computers and Education: Artificial Intelligence, 7*, 100269. [Crossref]

[6] Anderson, N., McGowan, A., Galway, L., Hanna, P., Collins, M., & Cutting, D. (2023, November). Implementing generative AI and large language models in education. In *2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS)* (pp. 1-6). IEEE. [Crossref]

[7] Boguslawski, S., Deer, R., & Dawson, M. G. (2025). Programming education and learner motivation in the age of generative AI: student and educator perspectives. *Information and Learning Sciences, 126*(1/2), 91-109. [Crossref]

[8] Allison, J., Hwang, G. J., Mayer, R. E., Pellas, N., Karnalim, O., de Freitas, S., ... & Sanusi, I. (2025). From generative AI to extended reality: Multidisciplinary perspectives on the challenges, opportunities, and future of educational computing. *Journal of Educational Computing Research, 63*(6), 1327-1363. [Crossref]

[9] Denny, P., Leinonen, J., Prather, J., Luxton-Reilly, A., Amarouche, T., Becker, B. A., & Reeves, B. N. (2024, March). Prompt Problems: A new programming exercise for the generative AI era. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (pp. 296-302). [Crossref]

[10] Wilkinson, G. G. (2024). Enhancing Generic Skills Development in Higher Education in the Era of Large Language Model Artificial Intelligence. *Journal of Higher Education Theory & Practice, 24*(3). [Crossref]

[11] Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education–where are the educators?. *International journal of educational technology in higher education, 16*(1), 1-27. [Crossref]

[12] Velázquez-García, L., Cedillo-Hernández, A., Colón-Vallejo, A., Del Pilar Longar-Blanco, M., & Cedillo-Hernández, M. (2025). AI-Based Applications Enhancing Computer Science Teaching in Higher Education. *Journal of Information Systems Engineering and Management, 10*(2), 14-32.

[13] Alanazi, M., Soh, B., Samra, H., & Li, A. (2025). The Influence of Artificial Intelligence Tools on Learning Outcomes in Computer Programming: A Systematic Review and Meta-Analysis. *Computers (2073-431X), 14*(5). [Crossref]

[14] Arya, V., Saraf, A., Chichkanov, N., Papa, A., & Romano, M. (2025). AI-enhanced competency transfer hubs: a conceptual framework for university-industry

engagement and knowledge sharing. *The Journal of Technology Transfer*, 1-31. [Crossref]

[15] Atchley, P., Pannell, H., Wofford, K., Hopkins, M., & Atchley, R. A. (2024). Human and AI collaboration in the higher education environment: opportunities and concerns. *Cognitive research: principles and implications*, *9*(1), 20. [Crossref]

Nan Guo received the B.E. degree from Beijing Forestry University, Beijing, China, in 2011, the M.E. degree from Beijing University of Technology, Beijing, China, in 2014, and the Ph.D. degree in Control Science and Engineering from Beijing University of Technology, Beijing, China, in 2022. Her research interests include deep learning, few-shot learning, intelligent modeling and optimal control of complex industrial systems, and air pollution prevention and control. (Email: guonan03@bjut.edu.cn)

Bing Han received the B.E. degree in automation from Southwest Petroleum University, Chengdu, Sichuan, China, in 2024. She is currently pursuing the M.E. degree in Control Engineering at Beijing University of Technology, Beijing, China. Her research interests include image generation, object detection, and model lightweighting. (Email: hanbing2024@emails.bjut.edu.cn)

Junfei Qiao is currently a professor at Beijing University of Technology. His research interests include intelligent modeling, optimization control, and artificial intelligence applications. (Email: junfeiq@bjut.edu.cn)