The Journal of Education Insights Vol 4 Issue 1 2026

Original Research Article

Open Access

Impact and Countermeasures of Generative AI on the Curriculum System of Computer Science in Universities

Dan-Yang Liu*

Daqing Open University, Daqing, Heilongjiang, 163311, China

***Correspondence to:** Dan-Yang Liu, Daqing Open University, Daqing, Heilongjiang, 163311, China, E-mail: ddanyang@qq.com

Abstract: Generative AI (GenAI), represented by Large Language Models (LLMs) and diffusion models, is rapidly reshaping the global technological landscape, with higher education—particularly computer science (CS) majors—at the forefront of this transformation. This paper systematically analyzes the multi-dimensional and deep-seated impact of GenAI on the curriculum system of university CS courses, spanning knowledge structures, competency cultivation, pedagogical paradigms, and ethical values. It first elucidates the technical characteristics of GenAI and its deconstructive effect on core course content such as traditional programming. Subsequently, it discusses the potential threat GenAI poses to students' foundational coding abilities while simultaneously enhancing their creativity. The paper then analyzes the necessitated shift in teaching modes from "knowledge transmission" to "competency guidance." Finally, a strategic framework for response is proposed, integrating "preservation of fundamentals" with "innovation." By reconstructing course objectives and other measures, the framework aims to strengthen human-machine collaboration, solidify ethical boundaries, and build a new ecosystem for future-oriented computer science talent cultivation.

Keywords: Generative AI; Higher Education; Computer Science Courses; Curriculum Reform; Human-Machine Collaboration; AI Ethics

Introduction

At the end of 2022, applications of Generative AI (GenAI), such as ChatGPT, triggered global attention. Unlike discriminative AI, GenAI possesses powerful capabilities in content creation and logical reasoning, evolving from a mere tool into a "collaborator" for creative work. This technological leap constitutes a fundamental challenge to the higher education system. As the core fortress of information technology, the computer science major in universities has a curriculum system built around the concept of "making machines perform tasks," covering a complete chain of knowledge. However, GenAI has significantly simplified and automated the process of "making machines work"; by using natural language to describe intentions, students can leverage AI to complete multiple tasks. While the concept of "intent as implementation" brings an efficiency revolution, it challenges the value foundations of traditional computer science education in cultivating students' computational thinking and other core abilities. In the face of technology-driven educational transformation, university computer science education can neither be conservative nor blindly follow trends. Generative AI acts as both a "catalyst" and a "reshaper," forcing us to reflect on fundamental questions. Therefore, systematically analyzing the impact and proposing response strategies is an urgent priority for the reform of computer science education in universities. This paper will conduct an in-depth exploration along the main line of "Impact—Reflection—Response."

1. Deconstruction and Impact of Generative AI on Traditional Computer Science Course Content

1.1 Impact on Programming Foundations and Software Engineering Courses

The core objective of traditional courses such as *Fundamentals of Programming* is to have students manually write code to solve problems, thereby internalizing knowledge such as programming paradigms. However, Generative AI (GenAI) can rapidly generate high-quality code snippets based on natural language prompts, allowing students to bypass arduous coding exercises and obtain "answers" directly. This brings a dual effect: on one hand, it lowers the entry barrier to programming, enabling students to see results faster and stimulating their interest; on the other hand, it triggers the risk of "competency outsourcing." Students may become overly dependent on AI, resulting in a superficial understanding of low-level details and the formation of a fragile knowledge structure, leaving them helpless when confronted with bugs in AI-generated code. Furthermore, the processes emphasized in *Software Engineering* courses are becoming partially automated with AI assistance. AI can automatically generate UML diagrams and other artifacts, making the tedious documentation and procedural specifications in traditional teaching seem redundant. However, this also obscures the importance of cultivating soft skills such as communication and collaboration within the software engineering lifecycle.

1.2 Impact on Algorithms and Theory Courses

Courses such as *Algorithm Design and Analysis* and *Theory of Computation* are the cornerstones of computer science, aiming to cultivate students' abilities in abstract modeling, complexity analysis, and formal proof/derivation. Although GenAI can provide implementations for common algorithms, it often fails to explain why an algorithm is effective, provide rigorous proofs of time/space complexity, or make optimal choices under specific constraints. More importantly, AI itself is based on statistics and pattern matching rather than strict logical deduction; it may generate "pseudo-proofs" that appear reasonable but are logically flawed [1]. This leads to a paradox: students may find it easier to obtain the "shell" of an algorithm (the code) but harder to reach its "soul" (the underlying thought). If the focus of these courses remains on the memorization and reproduction of classic algorithms, their value will be significantly diminished. Educators must consider how to use AI as an exploratory tool to guide students to question AI-generated answers and explore the mathematical beauty and engineering wisdom behind algorithms, rather than passively accepting results.

1.3 Impact on AI and Machine Learning Courses

Ironically, courses such as *AI* and *Machine Learning*—the very birthplaces of AI technology—have also been heavily impacted by Generative AI. Previously, these courses focused on teaching classic supervised/unsupervised learning models, neural network foundations, and optimization algorithms. Today, students can directly invoke powerful APIs (such as OpenAI's GPT series or Anthropic's Claude) to complete complex Natural Language Processing (NLP) and Computer Vision (CV) tasks without an in-depth understanding of the internal complexities of Transformer architectures, attention mechanisms, or the details of large-scale pre-training. This has led to the dilemma of "black-boxing" the curriculum. Students may become proficient AI "users" while lacking a profound understanding of how AI works or why it makes mistakes. Course content urgently needs to transition from "how to build models" to "how to understand, evaluate, guide, and govern models," incorporating topics such as the principles of Large Language Models (LLMs), prompt engineering, model hallucination, bias and fairness, and Explainable AI (XAI).

1.4 Impact on the Course Evaluation System

Traditional assessment methods, such as closed-book final exams, laboratory programming assignments, and course projects, are facing unprecedented challenges to their validity. Students can easily use Generative AI to complete assignments and projects, making it difficult for instructors to distinguish between a student's independent thinking and results generated by AI. This not only undermines academic integrity but also renders traditional evaluations meaningless as measures of actual student capability. The central challenge for educators now is how to design a new evaluation system that effectively assesses higher-order thinking skills, such as critical thinking, innovative design, and system integration.

2. Deep Reflection Behind the Impact: Repositioning Educational Goals

2.1 From "Coders" to "Problem Definers" and "Solution Architects"

Historically, computer science education focused largely on cultivating excellent "executors"—engineers capable of accurately translating requirements into code. However, in the era of Generative AI, execution-level tasks will be increasingly automated. The core competitiveness of future computer science talents will shift toward higher-level capabilities: accurately defining problems, deeply understanding user needs, designing system architectures under complex constraints, and critically evaluating and iteratively optimizing AI-generated solutions [2]. They are no longer mere "coders," but rather "problem definers," "solution architects," and "commanders of human-machine collaboration."

2.2 Expanding the Connotations of Computational Thinking

As the core literacy of computer science, the classic definition of computational thinking includes decomposition, pattern recognition, abstraction, and algorithm design. In the AI era, these connotations must be expanded. Beyond the aforementioned abilities, we must integrate metacognitive abilities (monitoring and regulating one's own and AI's thought processes), prompt engineering skills (how to effectively communicate with AI to achieve optimal results), uncertainty management (how to handle the ambiguity, uncertainty, and potential errors in AI outputs), and ethical speculation (how to responsibly use and deploy AI).

2.3 The Increasing Rather Than Diminishing Importance of Foundational Skills

A common misconception suggests that since AI can write code, foundational programming skills are no longer necessary. On the contrary, solid foundational skills are more important than ever. They are the prerequisite for students to effectively navigate AI, judge the quality of AI outputs, and intervene or repair systems when AI fails. An "AI user" without deep foundational knowledge is like a driver who holds the steering wheel without knowing the rules of the road—dangerous and inefficient. Therefore, the goal of education is not to abandon the basics, but to deeply integrate foundational training with AI application scenarios. This allows students to "learn through use" and "use while learning," creating a virtuous cycle.

3. Systematic Response Strategies: Constructing a New Future-Oriented Computer Science Curriculum Ecosystem

3.1 Reconstructing Course Objectives: Defining the Competency Map under "Human-Machine Collaboration"

The cultivation objectives of various courses require comprehensive revision. For example:

(1) *Fundamentals of Programming*: The goal should shift from "being able to independently write X lines of code" to "being able to efficiently generate prototypes using AI, while accurately understanding and debugging AI-generated code and mastering core programming concepts."

(2) *Software Engineering*: The objective should focus on "leading requirement analysis, architectural design, team collaboration, and risk management of complex software systems with AI assistance, ensuring system reliability, security, and maintainability."

(3) *AI*: The goal should be upgraded to "understanding the basic principles and limitations of Large Language Models (LLMs), mastering methods for prompt engineering, model fine-tuning, evaluation, and governance, and responsibly integrating AI into practical applications."

The entire curriculum system should collectively aim at cultivating a new generation of computer science talents possessing a "four-in-one" competency structure: Solid Foundations + Higher-Order Thinking + Human-Machine Collaboration + Ethical Responsibility.

3.2 Updating Teaching Content: Embracing AI and Integrating the Frontier

Course content must evolve with the times, proactively treating Generative AI as both a subject of study and a pedagogical tool.

(1) Developing New Courses: Introduce specialized courses such as *Principles and Applications of Generative AI, Prompt Engineering and Human-Computer Interaction, AI Ethics, Security and Governance,* and *Software Engineering Practice in the Era of Large Models*.

(2) Transforming Existing Courses: In *Database Systems*, incorporate Natural Language to SQL (NL2SQL) content; in Computer Networks, discuss AI applications in cybersecurity defense and attack; in *Operating Systems*, explore how AI optimizes resource scheduling [3].

(3) Pervasive AI Literacy: Integrate AI-related knowledge, skills, and ethical reflection as a "red thread" throughout all courses, forming a comprehensive, all-process, and multi-dimensional AI literacy education.

3.3 Innovating Teaching Methods: From "Lecture-Exercise" to "Inquiry-Co-creation"

The pedagogical model must shift from unidirectional knowledge indoctrination toward student-centered, inquiry-based, and project-based learning.

(1) Case-Based Learning (CBL): Introduce a wide range of authentic engineering cases containing AI elements to guide students in analyzing the roles, opportunities, and risks AI introduces.

(2) Adversarial Learning: Encourage students to "challenge" AI. For instance, students can intentionally provide AI with flawed requirements, observe the outputs, and perform corrections, or analyze security vulnerabilities in AI-generated code.

(3) Human-Machine Collaborative Projects: Design course projects that explicitly require students to collaborate with AI. However, the final report must detail the student's own thought process, the interaction history with the AI, a critical evaluation of the AI's output, and the key decisions and modifications made by the student. This not only exercises competency but also provides a basis for process-oriented evaluation.

3.4 Constructing a New Evaluation System: Focusing on Process and Higher-Order Abilities

The evaluation system acts as the "baton" guiding student behavior and must undergo fundamental reform.

(1) Emphasis on Process-Oriented Evaluation: Significantly increase the weight of process materials, such as classroom participation, group discussions, milestone reports, code review records, and AI interaction logs.

(2) Designing "AI-Immune" Assessment Tasks: Shift the focus of assessment to abilities that AI cannot easily replicate, such as open-ended system design, multi-scheme trade-off demonstrations, troubleshooting in complex scenarios, and the analysis of ethical dilemmas.

(3) Oral Defense and Demonstrations: Utilize face-to-face defenses to directly assess the depth of a student's understanding of their work and their capacity for independent thinking [4].

(4) Establishing New Academic Integrity Standards: Clearly define the circumstances under which AI can be used and the standards for citing AI-generated content, integrating these into the scope of academic ethics education.

3.5 Strengthening Faculty Development and Ethical Education

Teachers are the linchpin of this reform. Universities must invest resources in organizing faculty training to help educators master GenAI technologies, update their educational philosophies, and enhance their instructional design capabilities. Simultaneously, AI ethical education must be elevated to an unprecedented height. It is imperative that students deeply understand the risks associated with AI, such as bias, discrimination, privacy infringement, and misinformation. By cultivating their sense of social responsibility and professional ethics, we ensure that technological development remains steadfastly dedicated to human well-being.

Conclusion

Generative AI has exerted a massive impact on the curriculum system of university computer science courses, triggering a profound revolution in educational paradigms. It simultaneously challenges the content, methodology, and evaluation systems of traditional teaching while offering a historic opportunity to reconstruct talent cultivation models. In the face of such transformation, we must not retreat; instead, we must embrace it with an open mind, a forward-looking vision, and pragmatic action. The core of future computer science education lies in cultivating capabilities that AI does not possess: deep insight, rigorous criticism, systemic architecture, humanistic care, and ethical judgment. By reconstructing course objectives, updating content, innovating methodologies, establishing a new evaluation framework, and integrating ethical education, Generative AI can be transformed into an engine for educational innovation. This will allow us to cultivate exceptional computer science talents who can lead the technological era, master the new paradigm of human-machine collaboration, and create enduring social value. This is not only our educational responsibility but also the calling of our era.

References

[1] Qi, M. L. (2025). Teaching practice of computer courses based on generative artificial intelligence. Electronic Technology, 54(10), 390-391.

[2] Jiang, B. C., Song, J., Han, G. M., et al. (2025). Construction of computer network courses oriented towards generative artificial intelligence. Computer Education, (09), 43-47.

[3] Zhang, H. Z., Zhou, X. B., Xu, Y. H., et al. (2024). Innovation of computer programming course teaching empowered by generative artificial intelligence. Computer Education, (07), 44-48.

[4] Li, H. Q. (2025). Logical rationale and implementation strategy of interactive teaching for computer organization principles aided by generative artificial intelligence. China Education Technology & Equipment, (20), 52-54+60.