Construction and Research on the "Generative Inquiry" Teaching Model for Python Programming Course under the Empowerment of Artificial Intelligence Generated Content

Yu Fan*

Haikou University of Economics, Haikou, 571132, China
**Corresponding author: fy_808@126.com*

Abstract: *With the rapid advancement of generative artificial intelligence technology, the intelligent transformation of education continues to deepen. To address long-standing issues in the Python programming course, such as significant cognitive disparities among students and difficulties in teaching abstract concepts, this study constructs an AIGC-empowered "Generative Inquiry" teaching model based on constructivist theory. This model positions students as the main actors and AIGC as a cognitive partner, systematically reconstructing teaching objectives, the roles of teachers and students, teaching processes, and evaluation systems through a "generate-validate-iterate" inquiry process, thereby providing a comprehensive solution to overcome the limitations of traditional teaching. This research not only offers an innovative practical paradigm for integrating intelligent technology into disciplinary teaching but also opens new pathways for cultivating students' computational thinking and innovative abilities. Subsequent studies will involve interdisciplinary validation and empirical research on the model's effectiveness.*

Keywords: *AIGC; Generative Inquiry; Python Programming; Teaching Model; Computational Thinking*

Introduction

In the context of intelligent technology-driven educational transformation, the teaching paradigm is undergoing a transformation and upgrade from tool application to ecological restructuring. Artificial Intelligence Generated Content technology, with its powerful capabilities in contextual interaction and content generation, offers a new pathway to overcome the limitations of traditional education. The Python programming course, due to its emphasis on logic and practicality, places higher demands on teaching interaction and personalized support. However, existing teaching models exhibit significant shortcomings in addressing students' cognitive differences and teaching abstract concepts. By systematically integrating the core capabilities of AIGC with curriculum teaching requirements, this study constructs a "Generative Inquiry" teaching model centered on stimulating students' exploratory spirit and generative abilities. This exploration not only holds practical significance for advancing the reform of programming course teaching but also carries important value for enriching teaching theories in technology-enabled environments.

1. The Era Background and Theoretical Foundation of AIGC in Education

1.1 Evolution Trends of Educational Formats in the Intelligent Era

Educational formats are undergoing profound transformation alongside the iterative advancement of information technology. Digital transformation has become a core characteristic of global educational development, shifting its focus from the mere application of technological tools to the reshaping of teaching paradigms and learning ecosystems. This process manifests in the intellectualization of learning environments, the personalization of educational resources, and the precision of teaching management, aiming to construct a dynamic, open educational system capable of adapting to individual differences[1].

Against this backdrop, Artificial Intelligence Generated Content technology, as a key driving force, provides new possibilities for the evolution of educational formats. AIGC not only enhances the efficiency of distributing and presenting educational information but also, through its powerful content generation and interactive capabilities, promotes a shift in teaching structure from traditional knowledge transmission to stimulating student creativity and inquiry spirit. This new, generation-oriented educational ecology supported by intelligent technology constitutes the macro context for this study's exploration of the Generative Inquiry teaching model.

1.2 The Technical Core of AIGC and Its Potential for Educational Application

The technical core of AIGC is primarily built upon generative artificial intelligence models such as large language models and diffusion models. Through learning and training on massive datasets, these models acquire deep-level comprehension and generation capabilities, enabling them to produce high-quality multimodal content including text, code, and images based on given instructions or contexts. Its core technical features include contextual understanding, logical reasoning, and creative output.

This technical core determines its broad potential for application in the field of education. At the level of educational content, AIGC can dynamically generate exercises, case studies, and learning materials, achieving on-demand resource provision. At the level of the teaching process, it can assume the role of a personalized tutor, providing students with immediate feedback and heuristic prompts, thereby effectively reducing cognitive load. These potentials collectively point to a core value: AIGC can enable teachers to extricate themselves from repetitive tasks and focus more on instructional design and the cultivation of higher-order thinking skills.

1.3 New Developments in Constructivist and Inquiry-Based Learning Theory

Constructivist learning theory emphasizes that knowledge is actively constructed by learners through interaction with their environment, rather than being passively received. This theory posits that effective learning occurs in authentic, meaningful task contexts and is deepened through social collaboration. From this perspective, the teacher's role is that of a guide and facilitator, responsible for creating a learning environment conducive to meaning-making.

Inquiry-based learning theory is a significant practical pathway of constructivism, focusing on guiding students to acquire knowledge and develop scientific thinking by autonomously posing questions, seeking evidence, and drawing conclusions. The emergence of AIGC has injected new momentum into the development of these theories. It can simulate and construct complex, realistic problem scenarios, and act as an intelligent partner to engage in sustained dialogue with students, supporting them in proposing hypotheses and testing ideas. Consequently, the inquiry process is extended from the limited time and space of the classroom to a continuously accessible, resource-rich virtual inquiry community.

1.4 An Analysis of the Alignment Between AIGC and Programming Course Instruction

Programming courses, particularly Python language teaching, integrate a high degree of logic, practicality, and creativity. The traditional teaching model often faces challenges such as significant disparities in students' prior knowledge, difficulties in understanding abstract concepts, and insufficient resources for personalized tutoring. A notable tension exists between the intrinsic requirements of the course and its external constraints, urgently necessitating the introduction of new teaching philosophies and technical means to seek a breakthrough[2].

The characteristics of AIGC demonstrate a high degree of alignment with the needs of programming instruction. At the cognitive level, AIGC can translate abstract programming syntax and algorithmic thinking into natural language explanations and analogies, thereby lowering the barrier to entry for beginners. At the practical level, it can instantly generate code examples, debug error messages, and provide optimization suggestions, serving as an around-the-clock programming assistant. This deep interaction not only strengthens students' practical skills but also, through the "generate-validate-iterate" cycle, cultivates their computational thinking and problem-solving abilities in a subtle yet profound manner.

2. Composition and Implementation Path of the "Generative Inquiry" Teaching Model

2.1 The Core Connotation and Design Principles of the Generative Inquiry Teaching Model

2.1.1 The Core Connotation and Educational Value of the Model

The Generative Inquiry teaching model is a pedagogical paradigm, situated within the AIGC technology environment, that centers on students actively generating assessable learning outcomes as its core driving force. It transcends the level of viewing technology as a mere auxiliary tool, positioning AIGC as a cognitive partner capable of engaging in deep dialogue and collaboratively constructing knowledge with students. This model aims, through a spiraling process of "Generation-Validation-Iteration," to systematically cultivate students' computational thinking, algorithmic design, and innovative problem-solving abilities within the context of complex problem-solving in Python programming. Its educational value lies in achieving a shift from knowledge replication to knowledge creation.

2.1.2 The Guiding Principles for Model Construction

The construction of this model follows a set of interrelated design principles. The Subjectivity Principle ensures that students maintain the status of cognitive agents throughout their interaction with AIGC, where the role of technology is to empower rather than replace. The Generativity Principle requires that the output of all teaching activities must be concrete, reviewable, and iterative substantial outcomes, such as a segment of functional code, an optimized algorithm, or a solution document. The Scaffolding Principle emphasizes that AIGC should provide timely, appropriate prompts, feedback, and scaffolded support at key nodes in the student's cognitive process, thereby guiding them to bridge their zone of proximal development.

2.1.3 The Role Positioning of AIGC in the Model

Within this model, AIGC assumes multiple roles. It serves as a dynamic knowledge base, capable of instantly generating explanations, cases, and analogies related to specific programming concepts; it acts as an intelligent collaborator, participating in the process of code writing, debugging, and optimization; and it functions as an immediate feedback source, analyzing and evaluating the outcomes generated by students. This human-machine collaborative interaction redefines the multilateral interaction structure among teachers, students, and knowledge in the traditional classroom, constructing a more open and dynamic learning ecosystem[3].

2.2 Analysis of the Basic Components of the Teaching Model

2.2.1 Multidimensional Integrated Teaching Objective System

The teaching objective system of this model aims to achieve the deep integration of knowledge, skills, and literacy. At the knowledge level, it pursues a deep understanding and contextualized application of core concepts such as Python syntax, data structures, and algorithms. At the skill level, it focuses on cultivating computational thinking centered on problem decomposition, pattern recognition, abstract modeling, and algorithmic design, as well as the ability for efficient communication and iterative optimization in a human-machine collaborative environment. At the literacy level, it is committed to fostering students' critical thinking, a discerning attitude towards the use of technological tools, and the digital literacy required in the intelligent age.

2.2.2 Restructuring of Teacher-Student Roles and Interactive Relationships

The implementation of this teaching model leads to a fundamental restructuring of teacher and student roles. The teacher's responsibility shifts from being a one-way transmitter of knowledge to becoming a designer of learning scenarios, a planner of inquiry tasks, and a guide and facilitator of the human-machine collaborative process. Students, in turn, transform from passive knowledge recipients into active generative learners, initiators of exploratory actions, and questioners and evaluators engaged in deep dialogue with AIGC. This role transformation fosters a new type of teaching relationship centered around students' generative activities.

2.2.3 Generative Design of Teaching Content and Activities

The teaching content is organized into a series of challenging and open-ended generative task sequences, such as "designing a personalized recommendation algorithm" or "developing an automated data processing script." Teaching activities closely revolve around these tasks, manifesting specifically in the optimization of prompt engineering, the analysis and explanation of code generated by AIGC, the modular expansion of program functions, the collaborative debugging of complex bugs, and the refactoring and optimization of solutions. These activities collectively form a clear generative learning path.

2.3 Teaching Process Design Centered on Generation and Inquiry

2.3.1 Creation of Problem Scenarios and Task Analysis

The teaching process begins with a carefully designed, authentic programming problem scenario. This scenario aims to stimulate students' intrinsic motivation for inquiry and help them recognize the value of the learning task. At this stage, students need to independently or through group discussion analyze the complex problem, clarify requirement constraints, and form initial, informal solution ideas. This foundational work paves the way for effective subsequent interaction with AIGC, completing the preliminary mapping from the problem space to the solution space.

2.3.2 Human-Machine Collaborative Generation and Iteration Cycle

This is the core phase of the model. Students translate their initial ideas into precise prompts that AIGC can accurately understand, guiding it to generate initial code, algorithmic ideas, or reference cases. Subsequently, students are required to run, test, and critically examine the generated results, identifying logical errors, flaws in boundary conditions, or potential for optimization. Based on this analysis, students adjust or refine their prompts and interact with AIGC again to seek improvements, thereby entering a spiraling "generate-evaluate-regenerate" iterative cycle until a robust solution is produced[4].

2.3.3 Outcome Integration and Metacognitive Reflection

After obtaining satisfactory generated outcomes, the process moves into the integration and sublimation stage. Students need to integrate the AIGC-assisted, modularly generated code into a complete project and document its design rationale and iterative process. More importantly, it is essential to guide students in metacognitive reflection, reviewing the entire inquiry path, summarizing the strategic gains and losses in their collaboration with AIGC, their deeper understanding of programming concepts, and the growth in their problem-solving abilities. This achieves the consolidation and transfer of knowledge.

2.4 Support Conditions and Evaluation System for the Teaching Model

2.4.1 Key Technical Environment and Resource Preparation

The smooth operation of the model relies on a robust technical support environment. This includes reliable access to mainstream AIGC tool platforms, necessary computational resources, and a stable network environment. Regarding teaching resources, it is necessary to systematically develop a repository of generative task projects closely aligned with the Python course knowledge points. Each project should be equipped with clear learning objectives, difficulty grading, and reference prompt frameworks and evaluation criteria for teachers and students, ensuring the pedagogical effectiveness of the inquiry activities.

2.4.2 Evaluation Orientation Combining Process and Outcome

The evaluation system of this model adheres to an orientation that values both process and outcome equally. It breaks through the limitation of traditional evaluations that focus solely on the correctness of the final code, expanding the evaluative scope to encompass the entire inquiry process and the development of thinking. This means that the quality of students' interaction with AIGC, the problem-solving strategies demonstrated during the iterative process, the ability to critically evaluate generated content, and the innovativeness of the final code outcomes collectively form the core dimensions of the comprehensive evaluation.

2.4.3 Diversified Evaluation Methods and Tools

To achieve the aforementioned evaluation orientation, diversified evaluation methods are required. Process evaluation can rely on the analysis of AIGC dialogue logs, the review of historical records from code version control systems, and students' reflective reports on the process. Outcome evaluation, on the other hand, focuses on a comprehensive assessment of the final code's structural standardization, functional completeness, algorithmic efficiency, and the quality of the project documentation. This diversified evaluation system aims to provide a comprehensive and multi-dimensional profile of students' ability growth in the human-machine collaborative environment[5].

3. Theoretical Contributions and Research Prospects of the Teaching Model

3.1 The Theoretical Innovation Value of the Generative Inquiry Teaching Model

The Generative Inquiry teaching model provides a new practical paradigm for the development of constructivist learning theory in an intelligent technology environment. By defining AIGC as a cognitive partner and thinking scaffold, this model deepens the understanding of technology's role in education, transforming it from an auxiliary tool into an active agent in collaborative construction. This perspective expands the theoretical boundaries of human-machine collaborative learning and offers a fresh theoretical lens for understanding the social dimensions of knowledge construction in the intelligent era.

At the level of teaching model theory, this research systematically integrates the characteristics of generative artificial intelligence with the inquiry-based learning framework. Traditional inquiry models are often constrained by resource accessibility and feedback immediacy, whereas this model, through the generative capabilities of AIGC, creates a continuously available inquiry environment, realizing the teaching philosophy of "generation as inquiry." This integration not only enriches the theoretical system of technology-enhanced teaching models but also provides a new analytical dimension for research on the mechanisms of technology-enhanced learning.

3.2 Implications of This Research for Informatized Teaching Practice

This study provides an actionable reference path for the reform of programming course instruction. It demonstrates how to organically integrate AIGC throughout the entire teaching process, rather than through fragmented use, offering educators a comprehensive framework for redesigning courses in an intelligent technology environment. The model's emphasis on generative task design, human-machine dialogue strategies, and iterative evaluation methods holds direct reference value for the design of informatized teaching activities.

For teacher professional development, this research clarifies the specific directions for the transformation of the teacher's role in the intelligent era. Teachers need to transition from knowledge transmitters to designers of learning experiences and guides in human-machine collaboration. This transition requires teachers to master new instructional design capabilities and technology integration literacy. Simultaneously, the model's implication for educational technology integration is that successful technology application should focus on creating new types of cognitive activities, rather than simply replacing existing teaching components[6].

3.3 Limitations and Shortcomings of the Research

The scope of this study is confined to the specific disciplinary field of Python programming, and the applicability of the teaching model requires further validation across diverse disciplinary contexts. Differences in subject-specific knowledge structures and skill requirements may lead to variations in the model's implementation effectiveness, and this context dependency necessitates more thorough investigation in subsequent research.

Regarding the research methodology employed, this study primarily utilizes qualitative methods to construct a theoretical framework and lacks support from large-scale empirical data. The causal relationships among various elements within the teaching model, as well as the key variables influencing its effectiveness, have not yet been tested through rigorous experimental designs. Furthermore, the evaluation of learning outcomes predominantly focuses on short-term cognitive development, and tracking data on the model's impact on students' long-term competency retention and transfer remains insufficient.

3.4 Potential Directions for Future Research

Future research may dedicate efforts to the interdisciplinary transfer and adaptive adjustment of this model. Exploring its application potential and adaptation strategies in different disciplinary fields such as mathematical logic, scientific inquiry, or language creation will contribute to constructing a more universal theoretical framework. The matching mechanism between the characteristics of different disciplines and the capabilities of AIGC is a research direction worthy of in-depth exploration.

At the level of empirical research, there is a need to conduct strictly controlled teaching experiments to quantitatively analyze the model's impact on various levels of learning outcomes. Long-term tracking studies investigating the sustainable development of students' computational thinking and innovative abilities can provide more robust evidence for the model's effectiveness. Another important direction is research on optimizing the quality of AIGC interaction, which includes teaching strategies for prompt engineering, the formulation of evaluation standards for generated content, and pathways to enhance the efficiency of human-machine collaboration. These research endeavors will directly strengthen the model's operability and practical effectiveness.

Conclusion

This study systematically constructs an AIGC-empowered "Generative Inquiry" teaching model and elaborates on its theoretical foundation, component elements, implementation pathways, and support systems, thereby providing a comprehensive solution that integrates theory and practice for Python programming course instruction in the intelligent era. The theoretical contribution of this model lies in positioning AIGC as a cognitive partner, deepening the theoretical connotation of human-computer collaborative learning, and promoting the new development of the inquiry-based learning framework through the concept of "generation as inquiry." At the practical level, this study offers a clear action guide for frontline educators to systematically integrate AIGC technology and restructure teaching processes. Based on the limitations and shortcomings of this study, future research will focus on the transfer and adaptability of this model to other disciplinary fields, quantitatively evaluating and analyzing the long-term mechanisms through which it promotes student competency development via rigorously designed empirical studies. Concurrently, in-depth research on teaching strategies aimed at optimizing AIGC interaction quality will be conducted to continuously enhance the model's operability and practical effectiveness.

Fund Projects

Project supported by the Education Department of Hainan Province, project number: Hnjg2025ZC-97

References

[1] Zhao Guozhu, Zhao Huanhuan, and Gui Bao. "Research on the Dynamic Evaluation System of Learning Outcomes in AI-Empowered Python Programming Courses." Journal of Social Science of Jiamusi University 43.11 (2025): 165-168.
[2] Wang Xiaoye, et al. "An AIGC-Empowered Intelligent Code Evaluation System for Programming Courses." Fujian Computer 41.11 (2025): 81-88.
[3] Ge Qi. "An Empirical Study on the Learning Outcomes of AI Large Models Empowering 'Python Programming'." Plastic Packaging 35.05 (2025): 79-81.
[4] Chen Hong. "Research on the Application of Generative Artificial Intelligence in Secondary Vocational Python Programming Courses." Teacher 20 (2025): 146-148.
[5] Zhao Yadi, et al. "Exploration of Ideological and Political Education in Python Programming Courses Based on Diversified Teaching Design." Journal of Science, Education and Culture 18 (2025): 146-149.
[6] Wang Yiyuan. "Exploration of Teaching Practices in AIGC-Empowered Programming Courses at Vocational Schools." China Strategic Emerging Industries 27 (2025): 159-161.