Journal of International Education and Development
2026, VOL. 10, NO. 2, 7-11
DOI: 10.47297/wspiedWSP2516-250001.20261002

Generative AI-Driven Reform of Python Teaching in Art Colleges

Xiaomei Zhou

Nanjing University of Communication, Nanjing, Jiangsu, 211172, China

ABSTRACT
Under the dual background of "Art-Science Integration" and the construction of "New Liberal Arts," computer basic education in art colleges faces unprecedented opportunities for transformation and challenges. As a core bridge connecting artistic creativity with digital technology, the traditional teaching mode of Python is often limited by the thinking characteristics of art students, suffering from pain points such as high entry thresholds, weak learning motivation, and a disconnect between technology and application. The explosive development of Generative Artificial Intelligence (GenAI) provides new technical levers and cognitive tools to solve these problems. This paper deeply analyzes the current status of programming education in art colleges at home and abroad, pointing out core issues such as "code fear," "lack of context," and "singular evaluation." On this basis, it constructs a new "Human-AI Collaboration" teaching paradigm driven by GenAI and proposes systematic reform countermeasures ranging from curriculum content reorganization, teaching mode innovation, and evaluation system reshaping to ethical literacy cultivation. Research indicates that deeply integrating GenAI into Python teaching can not only significantly reduce the cognitive load of programming but also stimulate the creative potential of art students, promoting a fundamental leap in art education from "skill transmission" to "thinking empowerment."

KEYWORDS
Generative AI; Art colleges; Python teaching; Art-science integration; Human-AI collaboration; Computational thinking

1 Introduction

With the booming development of the digital economy and the iterative upgrading of artificial intelligence technologies, "Art-Science Integration" has become a significant trend in global higher education reform. Art is no longer merely an expression of sensibility; it is increasingly becoming a testing ground for interdisciplinary innovations such as data visualization, interaction design, and generative art. In this process, the Python language, with its concise syntax, powerful data processing capabilities, and rich creative programming libraries (e.g., Processing.py, p5py, TensorFlow), has become the preferred programming language for non-computer science majors in art colleges. However, art students generally possess active image-based thinking but relatively weak logical thinking. The traditional Python teaching mode, centered on syntax memorization and algorithm deduction, often leads to students developing a fear of difficulty, falling into the dilemma of "unable to understand, unable to write, and unable to apply." Since 2022, generative AI technologies represented by ChatGPT, Midjourney, and GitHub Copilot have risen rapidly, fundamentally changing the ways knowledge is acquired and code is generated.

2 Current Status of Research and Problem Analysis at Home and Abroad

2.1 Current Status of Foreign Research

In developed countries in Europe and America, computer education in art colleges started earlier and responds keenly to new technologies, undergoing a profound transformation from a "syntax-centric" to a "creativity-centric" approach. Long before the popularization of Generative AI, top institutions such as the MIT Media Lab and the Rhode Island School of Design (RISD) had already promoted the concept of "Creative Coding." They utilized Python libraries like Turtle and Pygame to focus on visual expression and interactive creation, allowing students to receive immediate feedback through "learning by doing." With the emergence of tools like Copilot, universities such as Stanford and Carnegie Mellon quickly adjusted their syllabi. They not only significantly reduced students' frustration caused by syntax errors by introducing AI assistants—enabling them to focus on logic design and creative realization—but also specifically opened modules on "Prompt Engineering for Code" to cultivate precise human-machine communication skills. This change has established a new paradigm of "Human-AI Collaboration." Foreign academia generally believes that the focus of education should shift from "hand-writing every line of code" to "evaluating, debugging, and optimizing AI-generated code." This transforms art students from mere executors into "curators" equipped with critical thinking and aesthetic judgment, better aligning with their disciplinary characteristics and future needs.

2.2 Current Status of Domestic Research

Although the reform of Python courses in Chinese art colleges has accelerated under the policy drive of "New Liberal Arts" and "New Arts," it remains in an exploratory stage. While many institutions have attempted to make up for students' shortcomings in digital technology by offering relevant courses, and some universities have tried to introduce Project-Based Learning (PBL) along with cases like data visualization and generative art, most courses still fail to break free from the stereotype of being compressed versions of science and engineering textbooks. They remain overly focused on syntax details such as variables and loops, lacking deep coupling with art professional scenarios. More critically, the domestic teaching community holds a cautious attitude towards Generative AI. Due to concerns about academic integrity, over-dependence, and "AI hallucinations," many teachers still restrict or prohibit its use in assignments. Consequently, research on how to systematically integrate Generative AI into the entire teaching process is scarce, and there is an urgent need to construct a mature localized teaching model to break through current bottlenecks.

2.3 Existing Major Problems

2.3.1 "Code Fear" and Cognitive Load OverloadArt students are accustomed to perceptual and intuitive thinking styles. When faced with abstract syntax symbols and rigorous logical structures, they tend to develop psychological resistance. In traditional teaching, a large amount of time is spent memorizing syntax and troubleshooting spelling errors, causing students to give up before experiencing the joy of programming. In the AI era, persisting with the requirement of "writing from scratch" is not only inefficient but also goes against the trend of technological development.

2.3.2 Disconnection Between Teaching Content and Professional ContextExisting Python course cases are mostly derived from science and engineering backgrounds (e. g., mathematical calculations, data processing tables), lacking relevance to professional scenarios in art design and media communication. Students struggle to understand "what is the use of learning this," leading to a lack of learning motivation. There is a shortage of course content that deeply integrates AI technology with the artistic creation process (such as idea divergence, sketch generation, and work iteration).

2.3.3 Singular and Lagging Evaluation SystemTraditional assessment methods mainly rely on final closed-book exams or standard code reproduction, focusing on the accuracy of syntax memorization. This evaluation method cannot measure students' computational thinking abilities, creative realization capabilities, or their ability to solve problems using AI tools. In the context where Generative AI can easily complete standard code tasks, the reliability and validity of the original evaluation system have significantly decreased.

3 Countermeasures for Teaching Reform Driven by Generative AI

Addressing the above problems, this paper proposes constructing a new paradigm of Generative AI-driven Python teaching reform characterized by "Human-AI Collaboration, Creativity Orientation, and Ethics First," unfolding across the following four dimensions.

3.1 Curriculum Content Reorganization: From "Syntax Memorization" to "Logic Construction and Creative Realization"

Against the backdrop of Generative AI significantly reducing the cost of code writing, curriculum content must decisively shift from "syntax memorization" to "logic construction and creative realization. " Firstly, mechanical explanations of low-level syntax details such as variable types should be weakened, while the cultivation of computational thinking—such as problem decomposition and abstraction—should be strengthened. AI should be utilized as a "real-time tutor" to guide students to understand logic through natural language interaction rather than rote memorization of code. Secondly, an exclusive "Art + Python" case library should be constructed. Customized cases covering image processing, spatial simulation, and interactive installations should be developed for different majors like Visual Communication and Environmental Design. An AI-assisted creation process of "Creative Conception → Prompt Generation of Framework → Manual Logic Optimization → Artistic Presentation" should be established. Finally, a core module on "Prompt Engineering" should be added to systematically teach students the skills of precisely describing requirements, setting constraints, and iteratively optimizing Prompts, treating this as a key literacy for future human-machine collaboration, thereby achieving a deep integration of technical learning and artistic innovation.

3.2 Teaching Mode Innovation: Constructing a "Dual-Teacher Classroom" and "Human-AI Collaborative" Learning Ecosystem

Breaking away from the traditional one-way indoctrination of "teacher lectures, students listen," a new teaching

ecosystem involving interaction among teachers, students, and AI should be established.

"Teacher + AI" Dual-Teacher Collaborative Instruction: The teacher's role shifts from a "knowledge transmitter" to a "learning designer" and "thinking guide." In the classroom, teachers are responsible for explaining core concepts, organizing logical frameworks, and commenting on creative schemes, while the AI assistant acts as a "pair programming partner," answering specific syntax questions, providing code examples, and debugging errors in real-time. Teachers can demonstrate how to use AI to rapidly prototype an idea, showcasing the efficient workflow of human-machine collaboration.

Implementation of "Reverse Teaching" and Code Review: Changing the "lecture first, practice later" mode, a "generate first, review later" reverse teaching approach is adopted. Teachers provide an artistic creation requirement, asking students to generate code using AI first, and then organize students into groups to conduct "Code Review" on the AI-generated code. Students need to identify logical loopholes, redundant parts, or unsafe factors and propose optimization plans. This mode effectively cultivates students' critical thinking and code appreciation abilities.

Full-Process AI Empowerment in Project-Based Learning (PBL): For final projects, students are required to complete a comprehensive "Generative Art" or "Data Visualization" project. The process includes: using AI for brainstorming and idea collection → using AI to generate technical feasibility schemes → human-machine collaborative coding and debugging → using AI to assist in writing project documentation and creative statements. The focus of assessment is on the students' ability to harness AI throughout the process, rather than solely on the originality of the code.

3.3 Reshaping the Evaluation System: From "Result-Oriented" to "Process and Ability Parity"

Adapting to the characteristics of the AI era, a diversified and process-oriented evaluation indicator system should be established.

Increasing Weight on "Prompt Quality" and "Interaction Process": Students' dialogue records with AI (Prompt History) should be included in the assessment scope. The precision of students' questioning, the logic of iterative optimization, and strategies for using AI to solve complex problems should be evaluated. Excellent Prompt design should be regarded as a high-level intellectual achievement.

Strengthening Assessment of "Code Review" and "Innovative Application": Reduce assessments on standard code reproduction and increase tests on the ability to modify, optimize, and innovatively apply AI-generated code. For example, provide a piece of AI-generated code with bugs and require students to fix it and expand new functions; or require students to integrate unique artistic styles or interactive logic based on AI-generated foundations.

Introducing Multi-dimensional Work Exhibition and Defense: Final evaluation should not only look at the code execution result but also consider the uniqueness of the work's creativity, the rationality of technical implementation, and the student's reflection on the creation process. During the defense, key questions should include: "Which parts of this project were completed by AI? Which were led by you? How did you solve hallucinations or errors produced by AI?" to verify the students' true mastery and subjectivity.

3.4 Ethical Literacy Cultivation: Fortifying the Value Bottom Line of "Technology for Good"

While enjoying the convenience of AI, it is crucial to attach great importance to ethical risks and incorporate them as an important part of ideological and political education in the curriculum.

Academic Integrity and Copyright Awareness Education: Clearly define the boundaries of AI use. Teach students that they must annotate when using AI to generate code and strictly prohibit directly plagiarizing AI works as personal originals. Deeply discuss the issue of copyright ownership of training data for Generative AI, guiding students to respect intellectual property rights in artistic creation and avoid legal risks.

Algorithmic Bias and Social Responsibility: Through case studies, reveal potential gender, racial, and other biases existing in AI models, as well as the misleading effects these biases may cause in artistic creation and data visualization. Cultivate students' critical awareness when using AI tools to ensure that technology applications conform to social ethics and public order.

Philosophical Reflection on Human-Machine Relations: Guide students to think about "What is the value of artists in the AI era?" Clarify that AI is a tool rather than a subject; the core of art lies in human emotions, thoughts, and unique life experiences. Encourage students to explore new aesthetics of human-machine collaboration rather than becoming appendages of AI.

4 Implementation Safeguards and Challenge Responses

4.1 Faculty Team Construction: Building a "Dual-Qualified" Cross-Border Teaching Team

Teachers are the core driving force for the implementation of educational reform. However, in the context of the rapid iteration of Generative AI, teachers in art colleges generally face technical anxiety and capability gaps. Firstly, a normalized mechanism for improving teachers' AI literacy must be established. Schools should organize special workshops to not only train teachers in the operational skills of mainstream large models but also deepen their understanding of the logical principles behind AI, ethical boundaries, and its infinite possibilities in artistic creation, transforming teachers from "technical bystanders" into "demonstrators of human-machine collaboration." Secondly, disciplinary barriers should be broken to construct a "Computer + Art" dual-tutor collaborative education mode. Encourage computer teachers to pair with art design teachers to jointly reconstruct curriculum syllabi and design interdisciplinary projects. Computer teachers ensure the rigor of technical logic, while art teachers control the aesthetics of creative expression; both sides explore the best combination point of technology and art through collision. Furthermore, external intellectual resources should be actively introduced by establishing close school-enterprise cooperation mechanisms. Invite industry experts from digital media, game development, and AIGC startups to serve as part-time tutors or deliver frontier lectures, bringing the latest AI application cases from the industry (such as Sora video generation, Stable Diffusion workflows, etc.) into the classroom to ensure teaching content does not lag behind industry development, thereby building a composite faculty team that understands both art laws and intelligent technologies.

4.2 Technical Platform Support: Constructing a Safe and Controllable Smart Teaching Ecosystem

Technical platforms are the physical foundation for integrating Generative AI into teaching. Art colleges need to focus on building infrastructure that is safe, stable, and adapted to professional needs. Given the risks of data leakage and uncontrollable content associated with public large models, schools should prioritize deploying privatized large models or purchasing education-specific API services. Through localized deployment, teaching data, student works, and creative concepts can be retained on internal school servers, strictly protecting intellectual property and privacy security while avoiding interference from public network instability in teaching progress. On this basis, an integrated teaching platform combining code writing, AI dialogue, and work exhibition should be built to achieve a seamless experience of "what is asked is obtained, and what is obtained is used." More importantly, the precipitation and sharing of teaching resources must be promoted. Schools should lead the establishment of school-level "Prompt Engineering Libraries" and "Art Code Case Libraries," encouraging teachers and students to upload and share excellent prompt templates, debugging experiences, and creative code snippets to form a dynamically growing knowledge ecosystem. This not only reduces repetitive labor but also allows lower-grade students to get started quickly by standing on the shoulders of predecessors. Meanwhile, the platform should possess intelligent learning analysis functions to record students' interaction trajectories with AI in real-time, providing data support for teachers to adjust teaching strategies, truly creating an open, safe, and efficient smart teaching ecosystem.

4.3 Challenge Response: Educational Wisdom to Turn Risks into Opportunities

Facing the challenges brought by Generative AI, we must not stop eating for fear of choking; instead, we should use educational wisdom to transform them into opportunities for deepening teaching reform. Regarding the concern of "students over-relying on AI," the key lies in reshaping the assessment orientation and task design. Teachers should reduce standardized code reproduction assignments and instead design high-order thinking tasks with high complexity, openness, and personalization. For example, requiring students to complete cross-modal creative integration projects or construct complex system architectures requiring multi-module collaboration. Such tasks often exceed the independent solving capabilities of current AI, forcing students to deeply engage in logic planning, parameter tuning, and result evaluation, thereby establishing the dominant position of humans in the "human-machine dance." Regarding "AI hallucinations" and code errors, they should not be viewed as a scourge but transformed into valuable teaching resources. Specifically set up "bug-hunting" sessions in the classroom to display AI-generated code that looks perfect but contains logical loopholes or security hazards, training students to act like "code detectives" to identify, criticize, and correct errors. This process not only cultivates students' rigorous scientific spirit and critical thinking but also enables them to deeply understand the limitations of AI, realizing that the value of human experts lies in the final check of results and ethical judgment, thereby achieving a leap in ability from "passive acceptance" to "active mastery."

5 Conclusion

The rise of Generative AI is not the end of Python teaching in art colleges but a profound paradigm revolution. It liberates us from tedious syntax memorization, allowing us to return to the essence of computing—the construction of logic and the expression of creativity. By reconstructing curriculum content, innovating teaching modes, reshaping evaluation systems, and adhering to ethical bottom lines, we can build a new teaching ecosystem of "Human-AI Collaboration and Infinite Creativity." In this transformation, Python is no longer just a programming language; it is a magic wand for art students to master intelligent technologies and expand the boundaries of creation. The goal of the reform is not to cultivate "coders" proficient in syntax, but to nurture a new generation of composite talents who possess computational thinking, know how to collaborate with humans and machines, and can tell artistic stories with technology. This is not only an inevitable choice for art colleges to respond to the challenges of the times but also the only way to promote the deepening of "Art-Science Integration" and empower the high-quality development of the cultural industry. In the future, with the further development of multimodal large models, the boundary between art and technology will become even more blurred. Our teaching reforms must also remain open and dynamic, continuously exploring the new landscape of art education in the era of human-machine symbiosis.

Funding

2025 Special Project for "Reform of General AI Courses and Basic Course Teaching" in Jiangsu Undergraduate Universities: Research on AI-Empowered Reform of Python Courses in Art Colleges under the Background of "Art-Science Integration" (NO: 2025ZNT-55)

About the Author

Xiaomei Zhou, female, Han ethnicity, born on October 26, 1980, in Nanjing, Jiangsu. She holds a Master's degree and serves as an Associate Professor at Nanjing University of Communication. Her research focus is on databases.

References

[1] Liu Jiani. Reform Strategies for the Teaching Mode of Art Design Courses in Higher Vocational Colleges under the Background of "Three Educations" Reform[J]. Shanghai Fashion & Costume, 2025, (01): 114-116.

[2] Guo Fanli. Research on the Reform Practice of Python Programming Course Teaching Oriented to Artificial Intelligence[J]. Information & Computer, 2026, 38(03): 198-201.

[3] Chen Nannan. Reconstruction of New Form Classroom Teaching Mode in Higher Vocational Colleges Based on Smart Teaching Tools—Taking Python Programming Course as an Example[J]. Information & Computer, 2026, 38(01): 231-234.

[4] Wang Yuqing, Li Rongrong, Li Mo, et al. Research on the Reform of Program Design Courses Deeply Empowered by AI—Taking Python Programming Course as an Example[J]. Shanxi Youth, 2025, (24): 118-120.

[5] Ding Yaonan. Exploration of Ideological and Political Teaching Reform of Python Programming Course under OBE Concept[J]. Standard Life, 2025, (09): 85-87.