



Exploring an AI-Enabled Digital Teaching Model Driven by Self-Directed Learning

Yingying Sun^a, Yanli Fu^{b*}, Di Zhou^c

Engineering University of PAP, Xian, China

^a39971976@qq.com, ^{b*}Corresponding author: 345765707@qq.com,
^c2497195389@qq.com

Abstract. This study addresses several challenges commonly encountered in programming education, including substantial differences in students' prior knowledge, insufficient support for self-directed learning, and difficulties in implementing effective process-oriented assessment. Taking the Fundamentals of Programming course as a case study, the study explores the construction of a digital teaching course model driven by artificial intelligence (AI) and self-directed learning. Relying on a SPOC-based online learning environment and a programming practice platform, learning process data and programming behavior data are collected to support the design of personalized learning pathways. During instructional implementation, a tiered, progressive competency-oriented structure and a blended learning approach are adopted to strengthen students' practical programming skills. In addition, a learning data-driven formative assessment mechanism is introduced to enable continuous monitoring and feedback of teaching effectiveness. The results of teaching practice indicate that the proposed model contributes to improved student engagement, higher quality completion of programming tasks, and enhanced overall competency development, thereby providing a viable reference for the digital transformation of programming-related courses.

Keywords: Artificial intelligence; autonomous learning; competency-based; digital teaching

1 Introduction

With the increasing integration of artificial intelligence and big data technologies into the educational domain, programming-related courses are gradually evolving toward digitalized and intelligent instructional models^[1]. However, traditional lecture-centered teaching approaches continue to exhibit notable limitations in addressing students' heterogeneous backgrounds and divergent learning paces. These limitations are particularly evident in military education contexts, where closer alignment between course content and position-oriented competency requirements remains a pressing challenge.

The Fundamentals of Programming course is characterized by strong practical orientation and a progressively structured knowledge system, which calls for

corresponding innovations in teaching design to more effectively cultivate students' programming skills and computational thinking abilities. Against this backdrop, this study explores the construction of a digital teaching model driven by artificial intelligence and self-directed learning. By collecting and analyzing learning behavior data, the proposed model supports the design of personalized learning pathways and the optimization of instructional processes, facilitating the integration of autonomous learning and precision-oriented teaching. Ultimately, this approach aims to enhance teaching effectiveness and students' comprehensive practical competencies, providing practical insights for the digital transformation of programming education.

2 Challenges in Programming Course Instruction

2.1 Insufficient Instructional Support for Differentiated Learning

In the instructional practice of programming-related courses, existing teaching organization strategies show limited adaptability to students' individual differences, and the principle of differentiated instruction remains difficult to implement effectively at the operational level. Classroom instruction is commonly organized around a uniform pace and standardized content, with limited provision for tiered instructional design and differentiated learning support. As a result, personalized learning approaches are relatively constrained, and self-directed learning is, to some extent, confined to a formal rather than substantive role. Within a single instructional setting, instructors face challenges in simultaneously addressing students with diverse prior knowledge levels and learning rhythms, leading to insufficient responsiveness to heterogeneous learning needs.

With respect to instructional content and pedagogical approaches, teaching practices tend to emphasize programming syntax and discrete knowledge points, while comparatively less attention is devoted to computational thinking development and integrated practical activities. The connection between programming knowledge and engineering-oriented practice therefore requires further strengthening. In addition, course assessment is still predominantly based on summative evaluation, with limited collection and analysis of learning process data. This constrains timely identification of students' learning status and, in turn, limits the effectiveness of personalized instructional regulation and differentiated teaching implementation.

2.2 Insufficient Data Support and Feedback Mechanisms for Self-Directed Learning

In the instructional practice of programming courses, students' self-directed learning processes lack systematic, end-to-end data support and feedback mechanisms. Existing learning platforms tend to analyze learning behaviors primarily at an aggregate level, making it difficult to accurately capture individual learning states and their dynamic changes over time. This limitation constrains instructors' ability to provide precise guidance during the learning process and reduces the effectiveness of data-informed decisions for personalized instruction. As learning scale and instructional complexity continue to increase, relying solely on manual approaches for learning guidance and

academic support has become increasingly insufficient. There is a growing need to introduce intelligent instructional support mechanisms grounded in learning behavior data analysis. Artificial intelligence technologies offer inherent advantages in monitoring learning processes, identifying learning states, and delivering personalized support, thereby providing a feasible pathway for establishing data-driven digital learning support systems oriented toward self-directed learning.

2.3 Insufficient Alignment Between Instructional Content and Practical Requirements, and Limited Efficiency in Resource Utilization

In the instructional practice of programming courses, a certain degree of misalignment persists between course content and position-oriented competency requirements. On the one hand, instructional design mechanisms for incorporating feedback from real-world tasks and evolving technological developments remain limited. Course content is typically updated on relatively long cycles, making it difficult to promptly reflect changes in information infrastructure development and application scenarios. As a result, gaps emerge between intended learning outcomes and practical competency requirements. On the other hand, the development and use of instructional resources are still largely characterized by decentralized management, with insufficient unified planning and coordination mechanisms. The level of resource sharing across instructional units remains limited, and high-quality cases and practical resources are not easily disseminated or reused across different levels or organizational contexts. These constraints reduce the overall effectiveness and utilization efficiency of instructional resources.

Taken together, the above analysis indicates that current programming course instruction exhibits structural limitations in instructional organization, support for self-directed learning, and coordination between instructional content and practical resources. Existing teaching models struggle to effectively accommodate students' differentiated learning needs or to provide sustained and precise support throughout the learning process. These challenges suggest that traditional instructional organization and manual intervention alone are no longer sufficient to meet the demands of competency-oriented and process-focused programming education. Consequently, there is a clear need to introduce data-driven and intelligent instructional improvement approaches. The development of digital teaching models supported by learning behavior data analysis offers a critical technical foundation for subsequent instructional design and implementation in programming education reform.

3 Technical Foundations and Instructional Design Framework

3.1 Educational Informatization and Digital Transformation

The theory of educational informatization emphasizes the transformative role of information technologies in reshaping educational models, aiming to overcome traditional temporal and spatial constraints and to enhance the accessibility and equity of education. Building upon this perspective, the concept of a digital learning ecosystem further conceptualizes learning environments as dynamic systems in which learners,

instructors, technologies, and data interact to support meaningful learning experiences^[3]. Aligned with this theoretical framework, the present study leverages artificial intelligence and big data technologies to explore the development of personalized and intelligent instructional models that support active knowledge construction in self-directed learning contexts. In addition, competency-oriented education theory highlights that educational objectives should prioritize the development of learners' abilities rather than the mere transmission of knowledge, with particular emphasis on practical applicability and skill development. When integrated into programming course instruction, this perspective provides a foundational rationale for instructional design, guiding students' progression from conceptual understanding to practical problem solving and comprehensive competency development.

3.2 Blended Learning Technologies

Blended learning technologies enhance instructional flexibility and diversity by integrating online and face-to-face learning activities^{[4][5]}. The combined use of SPOC (Small Private Online Course) and MOOC (Massive Open Online Course) platforms enables the provision of differentiated learning experiences. SPOC platforms support customized instructional delivery by allowing learning resources to be tailored to students' individual learning conditions, while MOOC platforms offer large-scale open resources that broaden learners' exposure and perspectives. On this basis, blended learning models facilitate the integration of instructor guidance and student self-directed learning, with in-class instruction and out-of-class practice complementing one another to form a coherent learning cycle. Such blended approaches not only strengthen students' self-directed learning capabilities but also enable more fine-grained and personalized learning support through technology-enhanced instructional design.

3.3 Artificial Intelligence Technologies

A review of the existing literature on AI-enabled digital teaching indicates a rapidly growing body of research in this field. In particular, publications have increased markedly since 2024, as illustrated in Figure 1. Current studies primarily focus on three major directions. First, research on the application of AI technologies in personalized learning explores how intelligent recommendation systems and learning analytics can be employed to support individualized educational experiences. Second, studies on AI-based learning behavior monitoring and assessment emphasize the use of data analytics to provide real-time feedback on learning processes and to dynamically adjust instructional content according to learners' progress. Third, research on the integration of AI technologies into adaptive learning systems investigates how algorithm-driven optimization of learning pathways can improve learning efficiency and effectiveness. Collectively, these studies demonstrate the considerable potential of AI-enabled digital teaching models in enhancing learning outcomes and supporting personalized instruction, as shown in Figure 2.

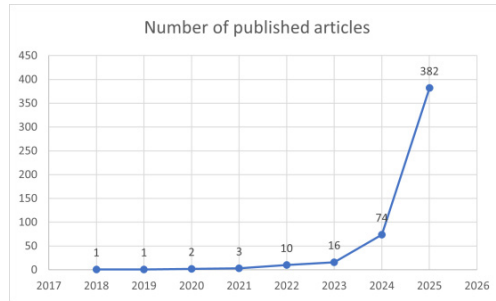


Fig. 1. Overall trend of the number of published literature

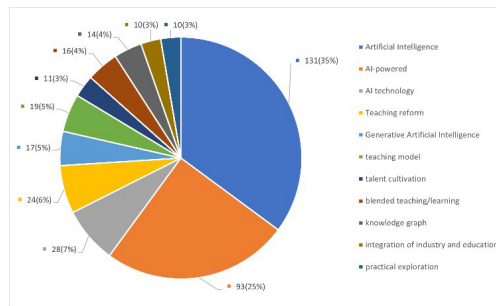


Fig. 2. ain Fields of Literature

Artificial intelligence technologies provide unprecedented support for educational practice, particularly in learning process monitoring and personalized feedback^[2]. Learning analytics techniques collect and analyze learners' activity data to construct individualized learning profiles, enabling instructors to gain timely insights into students' learning states and to implement data-informed instructional interventions. Adaptive learning systems leverage AI algorithms to dynamically adjust learning content based on learners' competency levels and learning progress, ensuring that instruction aligns with individual learning paces. Furthermore, the application of natural language processing and intelligent code analysis technologies enables AI systems to interpret students' programming inputs, diagnose errors, and offer targeted improvement suggestions, thereby substantially enhancing the interactivity and intelligence of programming instruction^{[6][7]}. These technologies provide essential data support and intelligent tools for building self-directed learning ecosystems, facilitating the design of personalized learning pathways and improving overall learning effectiveness.

3.4 Collection and Processing of Learning Behavior Data

To support AI-enabled instructional analysis and personalized learning support, this study systematically collected and processed students' learning behavior data during instructional practice, thereby establishing a stable and reliable data foundation for learning process analysis. Learning behavior data were primarily obtained from the

course SPOC platform and the programming practice platform. Data collection covered the entire learning process, including pre-class preparation, in-class learning, and post-class practice, enabling a comprehensive representation of students' learning engagement and practical performance.

With respect to data collection, this study focused on behavior indicators closely related to programming learning, which can be categorized into three groups. The first group comprises learning process behavior data, including video click frequency, duration of individual learning sessions, chapter completion rates, and temporal distributions of learning activities. These indicators were used to characterize students' learning engagement and learning pace. The second group consists of programming practice behavior data, such as the number of code submissions, compilation and runtime error counts, distributions of error types, and task completion time, which reflect students' operational proficiency and problem-solving abilities in programming practice. The third group includes stage-based learning outcome data, such as unit quiz scores, periodic assessment results, and completion status of practical tasks, which were used to provide a comprehensive evaluation of learning performance.

In terms of data processing, raw data automatically recorded by the platforms were first subjected to data cleaning and preprocessing procedures. These procedures included removing clearly abnormal records (e.g., prolonged login durations unrelated to active learning), merging duplicate submission records, and completing missing data caused by network or system issues. Subsequently, data from different sources and with different scales were normalized to ensure consistent statistical definitions of behavior indicators, thereby improving data comparability and stability. Through these processing steps, the integrity, consistency, and validity of learning behavior data were ensured to meet the requirements of subsequent instructional analysis and application.

Regarding data quality control, this study adopted a multi-source cross-validation strategy to verify learning behaviors. For example, learning duration data were examined in conjunction with code submission records to assess the authenticity of learning activities, thereby reducing potential biases associated with single indicators and enhancing the reliability of learner profile construction.

4 Exploring an AI-Enabled and Self-Directed Learning-Driven Digital Teaching Model

4.1 AI-Driven Reconstruction of the Instructional Model

To address the challenges identified in programming course instruction, this study proposes a personalized instructional model oriented toward individual differences by adopting the principles of “learning through practice” and “instruction informed by learning.” Emphasizing a competency-oriented and design-driven approach, the proposed model follows a progressive pathway from foundational knowledge to advanced application, and from knowledge acquisition to competency development. Specifically, the instructional model is organized into four interconnected modules: a basic programming module, an advanced programming module, a competition-oriented training

module, and a design-oriented practice module. Each module is associated with clearly defined instructional objectives and task arrangements, and the modules are hierarchically connected to support students' gradual progression based on their learning pace and competency levels.

1) The basic programming module focuses on the development of foundational programming skills, enabling students to acquire essential programming syntax and logical reasoning abilities. Through the integration of hands-on exercises and case-based activities, students move beyond isolated syntax learning toward initial applications of programming knowledge.

2) The advanced programming module introduces students to data structures, algorithms, and their practical applications, with an emphasis on cultivating the ability to solve more complex problems. Instruction at this stage places greater focus on programming thinking, requiring students to apply previously acquired knowledge to more challenging programming tasks that encourage exploration and innovation.

3) The competition-oriented training module incorporates programming contests and algorithmic challenges to provide students with opportunities to engage in problem solving under realistic constraints. Participation in competitive scenarios helps students enhance their analytical skills, adaptability, and performance under time pressure.

4) The design-oriented practice module, which constitutes the core component of the instructional model, emphasizes the development of comprehensive competencies through project-based design and implementation. Students are required to address authentic problem requirements by applying programming skills in engineering contexts, while simultaneously engaging in teamwork and project management activities to foster interdisciplinary capabilities and innovative thinking.

Through the hierarchical integration of these modules, the instructional model supports competency development at multiple levels while ensuring that students encounter appropriate challenges at each stage of learning. This staged design enables students to achieve sustained growth from foundational entry points to higher-level learning outcomes.

4.2 AI-Enabled Support for the Learning Process

Building upon the instructional model design, this study introduces artificial intelligence and big data analytics to provide intelligent support for the learning process, with particular emphasis on learning state identification and personalized learning pathway guidance. Considering the requirements for interpretability and practical implementability in instructional contexts, the construction of learning pathways does not rely on complex deep learning models. Instead, a rule-engine-based approach is adopted, integrating learning behavior analysis with pedagogically defined instructional rules.

In terms of learning state analysis, the system continuously collects and analyzes students' learning behavior data from the SPOC platform and the programming practice platform. Key indicators include code submission frequency, distributions of error types, learning duration, and task completion status. These behavioral indicators are mapped to specific knowledge units and competency objectives based on the course

knowledge structure, thereby forming stage-based learner profiles that characterize students' levels of knowledge mastery and practical competency development.

With respect to personalized learning pathway support, the system evaluates students' learning states according to predefined instructional rules and dynamically adjusts the recommendation of learning resources and practice tasks. For example, when a student exhibits persistently high error rates or low task completion efficiency within a particular knowledge unit, the system prioritizes the delivery of targeted explanatory materials and reinforcement exercises related to that unit. Conversely, when stable performance is observed and predefined thresholds are met, students are guided toward more advanced learning content or comprehensive practical tasks. These instructional rules are jointly defined based on course learning objectives and teaching experience, ensuring that pathway recommendations are grounded in clear pedagogical rationale and maintain strong interpretability.

Through the integration of rule-based decision mechanisms and learning behavior analysis, the proposed approach enables dynamic monitoring and tiered guidance throughout the learning process. This not only reduces instructors' workload in managing learning progress but also provides students with timely and targeted learning support, thereby forming a data-driven adaptive learning support loop, as illustrated in Figure 3.

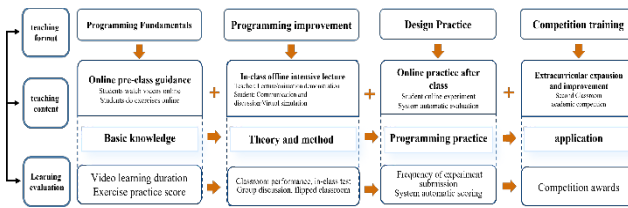


Fig. 3. Personalized Ability Cultivation and Autonomous Learning Ecosystem in Programming Courses

4.3 Development and Sharing of Digital Instructional Resources

This study places strong emphasis on the development of online resources and a digital course ecosystem to support programming instruction. Specific measures include the enhancement of supporting digital materials, such as MOOC- and SPOC-based course resources and concise programming tips, as well as the design of diversified programming experiments and practice tasks. These tasks encompass independent programming assignments, game-based challenge activities, competition-oriented training, and project-based design tasks, thereby balancing engagement with appropriate levels of challenge. In addition, a shared instructional case repository is constructed to support applied learning. The repository includes both foundational cases related to programming syntax and algorithms and practice-oriented project cases aligned with position-oriented competency requirements. Representative examples include data processing and task scheduling scenarios simulating information-centric operational environments, as well as cases involving data security and encryption algorithms in communication systems. These cases enhance the practical relevance of instructional content

while fostering student engagement and motivation. Through case-driven learning activities, students are able to strengthen their programming skills and integrated application capabilities, developing task-oriented problem-solving thinking. Furthermore, students are encouraged to conduct self-directed project-based research in small groups, which supports the development of collaboration skills and engineering thinking, and promotes deeper integration between in-class instruction and out-of-class learning activities.

To ensure the quality of instructional resources, this study introduces expert review and hierarchical management mechanisms during resource development. All online resources, cases, and project tasks undergo joint review by course instructors and domain experts prior to deployment, with particular attention to scientific validity, instructional appropriateness, and practical relevance. Structurally, instructional resources are organized into four hierarchical levels—foundation, enhancement, integration, and innovation—to ensure that learners at different proficiency levels receive appropriately matched learning support. In addition, the instructional platform incorporates a learning data-driven feedback mechanism to evaluate resource effectiveness. Based on indicators such as usage frequency, task completion outcomes, and learner evaluations, instructional resources are dynamically refined and updated, gradually forming a standardized and sustainable digital instructional resource system.

4.4 Learning Process Monitoring and Intelligent Assessment

To ensure the effective implementation of the proposed instructional model, a systematically designed assessment framework is established, emphasizing the integration of formative and summative assessment. Formative assessment encompasses multiple dimensions, including classroom participation, in-class quizzes, unit tests, post-class assignments, innovative tasks, and practical exercises, with particular attention paid to the collection of process-oriented data and the provision of staged feedback. Summative assessment primarily consists of comprehensive final projects and examinations, aiming to evaluate students' overall mastery of knowledge and their ability to apply programming skills in integrated contexts. In addition, AI technologies are employed to support real-time monitoring of students' learning processes and to generate personalized learning reports in a timely manner. These reports facilitate students' self-reflection while providing instructors with data-driven insights for adjusting instructional strategies. Through the establishment of this assessment feedback loop, the proposed approach mitigates the imbalance associated with outcome-focused evaluation and enables systematic monitoring and dynamic optimization of the entire learning process. The specific assessment dimensions and weightings are summarized in Table 1.

Table 1. Dimensions of Multi-process Evaluation for Programming Courses

	Evaluation element		proportion	evaluator	Evaluation method
Knowledge objective: 30%	Basic theoretical knowledge	The mastery level of core concepts,	12%	teacher	Classroom questioning, in-class quizzes, and homework

	Evaluation element		pro-portion	evalua-tor	Evaluation method
		principles, and basic methods			
	Expansion of professional knowledge	Frontier knowledge in relevant fields, and expanding understanding of content	9%	Teacher + learning group	Research project report or PPT presentation, group division Enjoy the task
	Knowledge application and comprehension	Can you apply the knowledge you have learned to the analysis of practical problems and understanding	9%	Teacher + learning group	Case analysis, classroom discussion
Ability target: 60%	Practical operation ability	In experimental and practical courses, can one be proficient and accurate Complete the specified operational tasks	24%	Teacher + teaching platform	Experimental report scoring, practical platform training
	Problem-solving ability	When facing practical problems, can one apply the knowledge and methods learned Adopt a systematic approach, analyze problems, propose solutions, and effectively implement them	18%	Teacher + teaching platform	Case practice, project practice, course design
	Team collaboration ability	Whether you can effectively communicate with team members, divide and cooperate tasks, and jointly complete project tasks	12%	Teacher + learning group	Using anonymous questionnaire
	innovation capability	During project practice, homework completion, and other processes, whether one can propose novel ideas, methods, or solutions	6%	teacher	During the evaluation of students' homework, project reports, classroom performance, and other aspects, teachers pay attention to students The innovative point
	learning attitude	Students' class attendance, classroom participation, and homework Completing tasks in a timely manner and with a high degree of diligence, etc	5%	Teacher + class representative	Class attendance records, classroom behavior observation, homework Submission status of assignments
Quality Goal: 10%	Communication and interaction skills	Can you articulate your viewpoints and thoughts clearly in class In group discussions and project	5%	teacher	Observe students' communication and interaction performance in classroom speaking, group

	Evaluation element		pro- por- tion	evalua- tor	Evaluation method
		presentations, can you communicate effectively with others and listen to their opinions			discussions, project presentations, and other activities

4.5 Analysis of Teaching Practice Effectiveness

To evaluate the effectiveness of the AI-enabled and self-directed learning-driven digital teaching model, this study conducted a comparative analysis using a specific session of the Fundamentals of Programming course. Two classes of the same grade level were selected for comparison: the experimental group adopted the proposed digital teaching model, while the control group followed the traditional instructional approach. In terms of academic performance, the experimental group showed a significant improvement in the average score of the comprehensive test administered at the end of the course, with a greater improvement compared to the control group. Regarding programming task completion rates, the experimental group demonstrated higher on-time completion rates and accuracy rates compared to the control group. Additionally, data from the learning behavior analytics platform revealed that students in the experimental group submitted more code and experienced a gradual reduction in low-level syntax errors, indicating improved programming proficiency and problem-solving abilities. In terms of learning process, students in the experimental group exhibited longer sustained learning durations and higher levels of task engagement, with a noticeable reduction in procrastination compared to the control group. Overall, the results suggest that the AI-enabled digital teaching model has a positive impact on enhancing learning outcomes, fostering greater learning engagement, and optimizing the learning process.

5 Conclusions

This study focuses on key challenges in programming course instruction, including difficulties in implementing differentiated instruction, insufficient utilization of instructional resources, limited support for students' self-directed learning, constraints of traditional teaching approaches, and weaknesses in assessment and feedback mechanisms. Based on an analysis of these issues, an AI-enabled and self-directed learning-oriented digital teaching model is proposed. By integrating online learning platforms, programming practice tasks, and learning data analytics, the proposed model seeks to overcome the limitations of traditional instructional approaches and to explore personalized and intelligent learning pathways. Through the application of learning data analysis, the model provides timely feedback and personalized instructional guidance, thereby enhancing overall teaching effectiveness. In addition, this study emphasizes the development of digital instructional resources, including online course platforms, interactive programming environments, and learning material repositories, which together offer comprehensive learning support for students. Furthermore, a learning data-driven

process monitoring and assessment framework is established, combining formative and summative assessment to support continuous learning and competency development through real-time feedback and individualized guidance. The results of teaching practice indicate that the AI-enabled digital teaching model contributes to improvements in students' programming skills and innovative thinking. At the same time, the model provides instructors with effective instructional support tools, facilitating data-informed teaching decisions and promoting innovation in instructional models. Overall, this study offers practical insights for advancing digital transformation and pedagogical innovation in programming education.

References

1. Su Xiaohong, Miao Qiguang, Chen Wenyu. Personalized Teaching Model Based on AI Empowerment and Industry-Education Integration to Enhance Programming Ability [J]. *Journal of Chinese University Teaching*, 2023, (06): 4-9
2. Zheng Haifeng. Research on the Reform of Digital Teaching Mode in Colleges and Universities Driven by Generative AI [J]. *Modern Commerce and Industry*, 2025, (17): 16-19. DOI: 10.19311/j.cnki.1672-3198.2025.17.006
3. Zhang Yuxin, Lin Shaozhen. Exploration and Reform Practice of Teaching Mode in Colleges and Universities under the Background of Digital Education [J]. *Journal of Heilongjiang Teachers Development University*, 2025, 44(07): 52-55. DOI: CNKI:SUN:HLJB.0.2025-07-011
4. Qin Fudian, Li Jing, Zhou Fukan. Research on the Practical Path of Digitalization Construction of Courses in the Context of Smart Education [J]. *Software Guide*, 2025, 24(07): 206-214
5. Liang Lili. Reform and Practice of Blended Teaching Mode for Programming Courses in the Context of Digital Education - Taking Java Programming Course as an Example [J]. *Information and Computer*, 2025, 37(02): 227-229. DOI: CNKI:SUN:XXDL.0.2025-02-056
6. Zhou Hao, Tan Qingfang. Practical Exploration of AIGC Technology Empowering Design Teaching Mode Innovation Driven by Industry-Education Integration [J]. *Printing and Digital Media Technology Research*, 2025, (S1): 25-29. DOI: 10.19370/j.cnki.cn10-1886/ts.2025.S1.005
7. Liu X .Research on Multiple Intelligence Teaching Mode of Digital Media Technology Major Driven by AI Big Model[J].*Journal of Commercial Biotechnology*,2025,30(3):1128-1137.DOI:10.5912/JCB2699.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

