International Journal of Innovation and Research in Educational Sciences
Volume 12, Issue 2, ISSN (Online) : 2349–5219

Teaching Reform of Principles and Applications of the Linux Operating System Course Based on Multi-Theory Integration and Artificial Intelligence

Lei Liu
School of Data Sciences, Zhejiang University of Finance and Economics, Hangzhou, China.
Corresponding author email id: leiliu@zufe.edu.cn

Date of publication (dd/mm/yyyy): 12/03/2025

Abstract **– This paper develops a framework by integrating constructivism, Bloom's Taxonomy, cognitive load theory, and feedback theory with artificial intelligence (AI) to meet the increasing demand for Linux expertise in data science education. We revamp the teaching methodology of the course Principles and Applications of the Linux Operating System by implementing a task-driven approach, layered learning, cognitive load reduction, timely feedback, and collaborative simulations of big data projects. Evaluation results exhibit substantial enhancements in students' practical skills, self-directed learning capabilities, and collaborative capacities. This framework offers instructors a systematic, scalable strategy to develop Linux proficiency aligned with the changing needs of data science professionals.**

Keywords **– Constructivism, Bloom's Taxonomy, Cognitive Load Theory, Feedback Theory, AI.**

I. Introduction

Big data technologies and data science are developing unprecedentedly in the digital era, fundamentally altering many industries and fields. Crucially fundamental for the major of data science and big data technologies, the "Principles and Applications of the Linux Operating System" course is also crucial for students' future professional growth. The Linux operating system is the platform for leading data processing technologies, including Hadoop and Spark.

Nevertheless, students' learning results and the development of their practical abilities are greatly influenced by the discrepancy between theory and practice, inadequate learning materials, and a lack of involvement and pragmatism in conventional teaching approaches. The frequency of these challenges highlights the need and urgency of reforming education.

Research on teaching reform now in publication lacks both theory and practice. For instance, some research lacks a teaching reform framework with multi-theory collaboration and concentrates only one or two theories, for example, adopting a teaching model based on UbD concept and constructivism in Java programming courses [1] or studying the impact of ChatGPT on students' cognitive load and learning outcomes [2]. Applying only one or two theories makes it challenging to improve students' comprehensive capacities fully and cannot help maximize the educational process. Practically, current studies also restrict the application of AI technologies. While AI is progressively being used in education, its application in Linux teaching is still in its infancy and lacks in-depth practical cases and rigorous study.

By including constructivism [3], Bloom's Taxonomy [4], cognitive load theory [5], and feedback theory [6], we seek to break the chains of conventional teaching models by deeply mixing them with AI to build a brand-new Linux teaching reform framework. Our framework will fully realize the benefits of every theory, depend on the strong features of AI, and ultimately maximize the functions of the "Principles and Applications of the Linux Operating System" course from many angles, including a task-driven approach, layered learning, cognitive load reduction, timely feedback, and AI-assisted instruction. Using this study, we aim to significantly improve students' practical skills, autonomous learning capacities, and teamwork skills; hence, we hope to give them a more methodical and practical course experience in data science and big data technology, lay a strong basis for their future professional development in the field of data science, and also provide valuable references and insights for teaching reform in related fields.

II. Theoretical Basis and Integration Advantages of Teaching Reform

1. *Constructivism*

Constructivism stresses the active building of knowledge since it holds that, instead of being passed on by teachers, knowledge is created by students in particular environments through social interaction and personal thought. Teaching strategies in many disciplines have been extensively used and confirmed under this idea [1, 7-11]. The Linux course is quite helpful in the education of data science and big data technology majors; constructivism offers an entirely appropriate theoretical framework for the instruction of this subject. Students must actively investigate knowledge, including file permission settings and directory layout design, by building learning environments similar to real-world working conditions, such as mimicking the building of a Data Storage and Access Control System when learning file management. Such a "learning by doing" method improves students' sense of involvement and exploratory energy and enables them to grasp functional operation skills rapidly and precisely.

2. *Bloom's Taxonomy*

Bloom's Taxonomy divides educational objectives from low to high into six levels: memory, understand--ing, application, analysis, synthesis, and evaluation, offering a clear framework for the setting of teaching content and evaluation criteria in many disciplines [12-14]. In Linux courses, we can help students advance from fundamental operating memory and knowledge to applying and synthesizing challenging projects. First, students must remember fundamental commands and operation terminology and understand their functions and application scenarios; then, using designing programming tasks and system configuration practices, progressively increase the difficulty of tasks, guiding students to apply the knowledge they have learned to practical projects and complete complex tasks such as automated script writing and distributed system building. By providing explicit learning objectives at several phases, this layered teaching and evaluation approach helps students progressively raise their Linux operational capacity.

3. *Cognitive Load Theory*

Particularly when learning complex ideas and abilities, Cognitive Load Theory underlines that teaching design should reduce students' cognitive load [2, 15-16]. Students in Linux classes must deal with many instructions and complicated system procedures, which can readily cause a cognitive load that is too high and influence learning results. To lower the learning difficulty, we can teach complex knowledge and operations progressively to students using segmented teaching and task breakdown. For instance, when teaching Shell script programming, we can start with simple variable assignments and conditional statements so that students may master the fundamental syntax, then introduce gradually more complex ideas such as loop statements and functions. In this process, we can design small programming tasks to improve students' programming abilities in practice progressively. Simultaneously, during the teaching process, we can also employ group projects to let students support one another and learn together, therefore lowering the cognitive load throughout instruction. Based on cognitive load theory, this teaching approach enables students to acquire Linux knowledge and abilities in a more laid-back environment and increases learning effectiveness by using more freedom.

4. *Feedback Theory*

The learning process depends on feedback in a significant part. By objectively perceiving their learning condition, feedback can assist students in understanding their degree of learning and reduce the pressure and stress resulting from high-difficulty learning goals [6]. Many researchers have paid close attention to feedback since it is a fundamental tool for applying successful teaching [17-18]. Rich and varied are the application forms of feedback used in Linux operation training. On the one hand, during experimental operations, we can assist students in grasping the proper operation process and methodologies, observe students' operation processes in real-time, give quick feedback and correction for students' erroneous operations, and so aid them. Conversely, we can provide students with more thorough and objective comments using technological tools, including automated feedback systems and peer review. Through peer review and conversation, students can also exchange their experiences, insights, and ideas for improvement with one another. This diverse feedback system helps students find and fix their errors quickly. It improves their reflecting skills and awareness of autonomous learning, supporting their in-depth knowledge and mastery of Linux.

5. *The Impact of AI on Education*

AI applications in education [19-23] present unheard-of opportunities to progress education's intelligence, personalization, and precision. AI can efficiently process a lot of educational data and generate exact student situation analysis models to grasp each student's learning traits and knowledge mastery using machine learning, deep learning, and other algorithms, providing a scientific basis for teaching students according to their aptitude and personalized learning. Concurrent with this, AI might help reduce the effort of lesson preparation, maximize teaching efficiency and quality using intelligent recommendation algorithms, provide teachers with tailored teaching aids and teaching plan references, and improve teaching design.

6. *Advantages of Theory Integration*

This study integrates constructivism, Bloom's Taxonomy, cognitive load theory, and feedback theory with contemporary research to develop a comprehensive and collaborative education reform framework guided by each theoretical approach. This new theoretical integration's combined effect is more than the sum of its components. Bloom's Taxonomy guides students to progressively understand Linux knowledge and abilities from essential to advanced, offering an obvious learning route and goal hierarchy for course design and evaluation. Constructivism fosters students' active learning, stresses their primary responsibility in educati--on, and generates knowledge using cooperative learning and active exploration. Cognitive load theory helps students acquire and master knowledge more quickly, improving the teaching process. Feedback theory guarantees that students can quickly modify their course of study and raise their academic performance.

These integrated theories complement each other and, by their combined optimization of efficacy, assist in maximizing the educational process. Moreover, integrating AI injects significant power into teaching reform through intelligent tutoring systems, individualized learning path planning, and automatic evaluation and feedback, thus boosting the intelligence and customizing of teaching. In Fig. 1, we show the integrated teaching reform framework for Linux education.

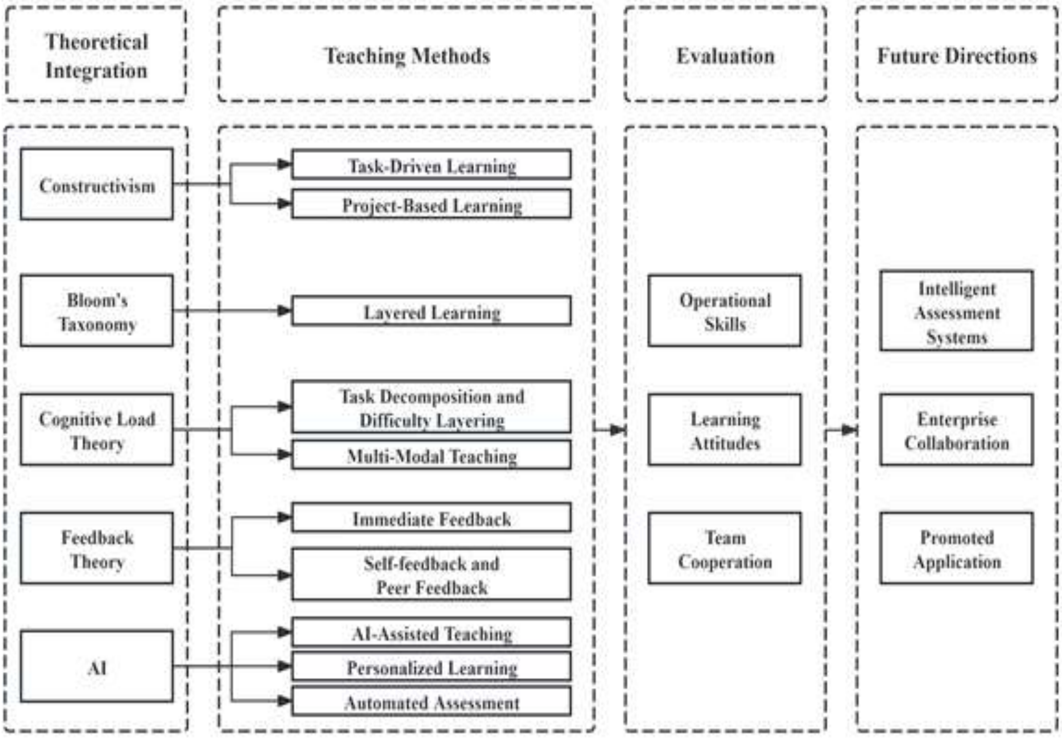

Fig. 1. Integrated Teaching Reform Framework for Linux Education.

III. Linux Course Design Based on Constructivism

1. *Task-Driven Learning*

Task-driven learning is one of the most effective teaching methods in constructivist education. Well-crafted task environments and guidance can enhance students' motivation and interest in learning, helping them construct knowledge as they complete tasks. We can design scenarios that are closely related to data science, such as simulating data server configuration, database management, data analysis, etc., to enable students to deeply experience the actual application scenarios of the Linux system in the process of completing tasks, deepening their knowledge of Linux knowledge and skills. In the chapter on File Management, for instance, we can create a task of "Creating a Secure Data File Sharing Environment," which calls for students to accomplish effective management and secure sharing of data files using particular operations, including file permissions and shared directory configuration. Students must utilize the knowledge and skills of file management they have acquired-file permission settings for read, write, and execute-as well as user and group administration to finish the job. In this regard, students can not only master the basic knowledge of file management but also grasp the practical application value of this information in data science, broadening their comprehension of the Linux system and increasing their desire for learning and initiative.

2. *Project-Based Learning*

Project-based learning is a teaching method that enables students to acquire thorough capability and functional abilities. Project-based learning in Linux courses can help students significantly boost their hands-on experience, problem-solving ability, and capacity for group projects. We provide actual project duties in data science, such as data cleansing and data processing pipeline building for each learning group. Students must apply their Linux skills, such as writing Shell scripts, setting system environments, installing and using data analysis tools, etc., to help them fulfill their project duties during project implementation. Using continuous practice and research under the direction of their projects, students can progressively become familiar with the operation instructions and Linux system operations and grasp the tactics and ideas for managing challenging circumstances.

Additionally, letting students learn and collaborate as a team is a teaching mode of project-based learning. Through group projects and discussions, students can share their thoughts and experiences, develop from one another, and progress collectively to accomplish knowledge sharing and extension. Under the comprehensive scenario of the Docker Usage Foundation chapter, for example, students can work together to establish a Docker data analysis environment. Following their joint study of the basic ideas and concepts of Docker, including the benefits of containerization technology and the link between images and containers, group members divide the work and cooperate to complete tasks, including Docker image construction, container operation, and configuration, so realizing the building of the data analysis environment. Using this approach, students can improve their Linux operating performance and acquire excellent communication and teamwork abilities, providing a good foundation for following fieldwork in data science.

3. *Professional Specialty Cases*

Linux supports general data science applications, such as the Hadoop cluster. Students require these unique examples to study Linux and data science. Hadoop clusters require Linux installation and a multi-node architecture for distributed data storage and computation. Students can learn Linux system manageme--nt, network configuration, software deployment, and Hadoop cluster operations and configuration through hands-on experience. Firstly, we introduce Hadoop's distributed file system (HDFS) and MapReduce computing architecture, as well as how to design Hadoop clusters, and then assist students in installing and configuring Hadoop software on Linux after they have configured HDFS, MapReduce, and Hadoop environmental variables. Students may experience issues with network connectivity and permissions setup during this process. Teacher consultations and class discussions inspire students to actively seek solutions and solve problems independently, improving their autonomous learning and problem-solving abilities. Students may master Linux system network management, permission management, and the fundamentals of distributed data storage and processing by creating Hadoop clusters, which will assist them in handling significant data chores in data science.

IV. Learning Level Design Guided by Bloom's Taxonomy

1. *Layered Learning Content Design*

Bloom's Taxonomy indicates that the content of the Linux course can be organized into three levels: me-

-mory, understanding, and application. After acquiring the fundamental information, students can progressi--vely graduate to the advanced application stage.

In the " memory " stage, students can concentrate on learning the fundamental Linux system commands and terminology, such as regular file operation commands (cp, mv, rm, etc.) and user management procedures (useradd, userdel, etc.). They can acquire a decent recollection of these commands and their underlying meanings using constant repetition and recall. With classroom lectures, operating models, activities, etc., we can assist students in combining memories.

In the "understanding" stage, we can help students study the principles and application scenarios behind these commands, including how process control achieves job scheduling and management and how the file management system handles the storage and access of files and directories. Clarifying the Linux system's core mechanics and practical application scenarios helps students be utterly aware of commands and ideas. When teaching file management, for example, teachers can demonstrate to students how the file system arranges and stores files, how file rights control file access permissions, and how file sharing helps several users access files.

Students can complete several helpful tasks in the "application" stage, including automated script development, system configuration, and optimization. Teachers can design large-scale projects where students must use their newly gained information and skills to handle practical problems. For example, students can maximize Linux server configuration to increase system stability and performance or design a Shell script to back up automatically and clean system logs. This slow learning content design not only increases the enjoyment of learning but also efficiently improves students' operational skills, enabling students to apply their knowledge to practical work better and provide strong technical support for the development of data science and big data technologies majors.

2. *Ability Assessment Based on Goals*

We can use corresponding assessment strategies at several learning levels to evaluate students' knowledge at every stage of their education. Objective question types like multiple-choice questions, fill-in-the-blank questions, and term explanations allow us to assess students' "memory" level recall of basic facts. For example, Multiple-choice tests can be created to assess students' knowledge of Linux common commands and their purposes.

Short answers, discussion questions, and case analysis are among the "understanding" questions used to measure students' knowledge-application skills. For instance, students can describe concepts and application scenarios for a certain Linux-related topic.

Plan modest project tasks at the "application" level, requiring students to use their newly gained abilities to address practical issues. These projects include setting up a distributed system, developing a management script, etc.

Based on objectives, this ability evaluation approach guides students toward well-defined learning goals at every level. Assessment reinforces the acquired knowledge, enhances Linux's operational capacity, and provides a solid foundation for the research and activity of significant fields in data science and big data technologies.

V. Teaching Design and Strategies Applying Cognitive Load Theory

1. *Task Decomposition and Difficulty Layering*

The course materials should be broken out based on difficulty to lessen the cognitive load for the students. Before progressively presenting more advanced material, such as permission management and script creation, we should ensure students completely master basic commands and simple procedures. We can organize assignments in laboratory classrooms by module, allowing students to work methodically and adapt to the operational challenges at each stage. In the broad sense of user and group administration, for instance, the duties of the data analysis team can be separated into subtasks, including user creation, user group creation, and user permission setting. Once they finish each assignment, students can progressively develop an overall awareness of user and group management, lowering the cognitive load when handling challenging user management chores. Furthermore, applying a small-step progressive teaching approach to provide knowledge and skills to students progressively helps them build a knowledge system more efficiently and enhance learning outcomes, providing a strong theoretical and practical basis for managing challenging user permissions and data access control issues in data science work.

2. *Use of Multi-Modal Teaching Resources*

Tools for multi-modal instruction can assist students in reducing their cognitive load. We can use step-by-step demonstration teaching strategies to progressively show the operation process through videos and explanations for more challenging operations to help students grasp each phase's meaning and operation points more precisely. Concurrently, colorful learning materials are helpful for students to review after class and enhance their memory of the course content. In the comprehensive situation of text processing and regular expressions, for example, complete charts can describe the matching rules and application scenarios of regular expressions, helping students better grasp and acquire the relevant knowledge and abilities. Showing how to apply regular expressions for log file analysis and processing can be done with demonstration movies. Using multi-modal teaching resources and strong support for text data analysis and processing in data science helps improve teaching quality and learning outcomes by reducing the psychological burden on students during the learning process and enhancing the presentation of abstract ideas and complicated processes.

VI. Application Strategies of Feedback in Linux Courses

1. *Application of Immediate Feedback*

Correcting operational teaching faults depends on teachers' quick responses. When students enter commands and solve spelling or operational problems, we should give directions immediately to enable them to grasp the proper technique. When students are learning disk storage management, for example, we can immediately comment on the mistakes and shortcomings in the disk partitioning and mounting management scripts produced by students, including syntax errors and logical loopholes, and offer improvement suggestions, optimizing the stability and dependability of the script. This rapid feedback system lets students pick knowledge and abilities fast and helps remove uncertainty during the operating process. Moreover, fast responses to problem identification inspire students more and help to create a good learning environment.

2. *Self-Feedback and Peer Feedback*

Promoting students' autonomous learning awareness and reflective skills depends much on peer and self-feedback. Using self-assessment forms to help students document their learning process and the challenges faced, we can guide students toward self-assessment in the classroom so that they may reflect on the following class. When writing Shell scripts, for instance, students can document the mistakes they find and the fixes, as well as the techniques and successes in finishing project activities.

Simultaneously, peer feedback has a significant impact on Linux instruction. Through group discussions and reciprocal evaluation, students can exchange their educational encounters and recommend areas for development. When constructing dispersed cluster networks, for instance, students can share configuration knowledge and optimization recommendations so network management capabilities can be collaboratively enhanced through improved network topology and network transmission speed.

This self-feedback and peer feedback system improves the learning environment and helps students develop their teamwork skills. It enables them to consolidate their knowledge of Linux in cooperation and raise their general technical level, thus laying a reasonable basis for the future execution of team projects in data science.

VII. Application of AI in Linux Course Teaching

1. *Intelligent Tutoring System*

Linux education can benefit from an automated tutoring system that uses learning data and behavior patterns to provide personalized instruction. When students struggle with Linux commands, the system can check their operation records and error types to assess their understanding and problems. It will then suggest relevant learning materials, video tutorials, or similar scenarios to assist users in grasping the commands. Depending on their mastery and learning progress, the system may also develop customized workouts and exam questions for each student, helping them practice and improve learning results. When students learn system services and startup management, the intelligent tutoring system can push relevant learning resources, such as detailed explanations of systemd service configuration files and timed task scheduling cases, and create exercises to test the student's systemd knowledge and application. Data science and big data technology majors need substantial assistance for ongoing learning and skill development to master Linux and improve their autonomous learning. Thus, the intelligent tutoring system can quickly alter learning tactics based on student comments and outcomes to provide more exact learning support.

2. *Personalized Learning Path Planning*

AI data analysis lets students choose a learning path based on their learning styles, knowledge, and goals. Linux training may require different skills and interests from students. AI may analyze students' learning data-including online learning time, exercise completion, and exam results-to assess their knowledge and learning preferences and offer relevant learning materials and sequences.

Relevant courses and practical projects, such as server security reinforcement and system performance o-

-ptimization, can be recommended for students interested in system management. Relevant data processing tools and data analysis cases can also be recommended for students interested in data processing and analysis. These can help students apply their strengths and interests in personalized learning paths and boost their learning enthusiasm and initiative. This customized learning path design helps students learn Linux more confidently and efficiently by recognizing and meeting their specific needs.

3. *Automatic Assessment and Feedback*

Most Linux courses' evaluation and feedback are AI-driven. Automatic assessment systems precisely evaluate students' homework and project results, providing timely feedback. After Linux system configurati--on or programming projects, the automatic assessment system can review students' code, configuration files, etc., find flaws, and provide complete feedback, including recommendations on system configuration security and stability, code readability, standardization, and execution efficiency. The automatic evaluation system may also generate learning reports based on student comments, helping students understand their knowledge growth and highlight areas for improvement, allowing them to modify learning methods quickly. After building the Hadoop 3.4.0 pseudo-distributed Docker image, the automated assessment system can evaluate the results, identify issues with file system configuration, Hadoop service status, and network connection settings, and make detailed recommendations for improvement, such as optimizing the Docker image size and adjusting Hadoop configuration file parameters, to help users quickly fix and improve the image.

AI technology improves assessment accuracy and efficiency, providing students with more detailed learning feedback. It encourages independent learning and skill development and gives examples of data science and big data technology majors' assessment and feedback innovations.

VIII. Evaluation of the Effectiveness of Teaching Reform and Outlook

1. *Evaluation of the Effectiveness of Teaching Reform*

The effectiveness of reform education depends on students' operational skills, learning attitudes, and teamwork. In the Linux course, task-driven and project-based learning has converted students' mastery of "mechanical memory" to "active exploration," enhancing their command operations abilities. Also, the average project score has increased, experimental completion has increased, and code error has decreased. Students' operational abilities have grown greatly since the educational reform, allowing them to do Linux operational tasks more accurately. The teaching reform also transformed students' learning patterns from passive to active, increasing their interest in Linux and autonomous study time. Teamwork skills like task decomposition, resource sharing, and task distribution have improved with group projects.

According to surveys, students enjoy this teaching method, and it has increased their knowledge of Linux in real-world data science projects and helped them learn Linux operations.

2. *Future Directions for Reform*

While this teaching reform has produced notable outcomes, development is still needed. Future Linux education should consider including more intelligent evaluation systems and adaptive learning tools to evaluate students' performance more precisely during the learning process. By automatically generating tailored learning recommendations and operation assistance based on student learning data, intelligent assessment systems help increase teaching efficiency and efficacy. Furthermore, included in the content of Linux courses are internship projects in collaboration with businesses, allowing students to apply the knowledge they have acquired in actual working conditions and enabling better meeting of future work expectations.

Moreover, depending on the experience of this teaching reform, similar reform approaches can be encouraged in other computer technology courses in the future to improve students' practical skills and learning excitement in many technological courses.

IX. Conclusion

By applying constructivism, Bloom's Taxonomy, cognitive load theory, and feedback theory, this study develops a framework for teaching reform. It extensively integrates AI to design a set of teaching tactics relevant to the Linux course, obtaining notable teaching results. The practical application of the teaching reform gives fresh ideas and approaches to Linux teaching. It is a valuable reference for the practical teaching reform of other computer skill courses. To provide students with a complete learning support system and help them develop outstanding talents in the field of data science, we should continue to explore areas including intelligent assessment, enterprise cooperation, and promoted application in the future.

Acknowledgments

This research was supported by the First Class Discipline of Zhejiang- A (Zhejiang University of Finance and Economics - Statistics).

References

[1] X. Yu, Z. Li, P. Gao, et al., "Java Programming Teaching Practice Based on UbD Concept and Constructivism Theory," *Compu--ter Education*, vol. 2, no. 2, 2023, pp. 54-57. (In Chinese)
[2] L.P. Patac and A.V. Patac Jr, "Using ChatGPT for academic support: Managing cognitive load and enhancing learning efficiency –A phenomenological approach," *Social Sciences & Humanities Open*, vol. 11, 2025, p. 101301.
[3] P. Wen and G. Jia, "Constructivism Theory and Teaching Reform-A Review of Constructivist Learning Theory," *Theory and Practice of Education*, vol. 22, no. 5, 2002, p. 6. (In Chinese)
[4] B.S. Bloom, *Taxonomy of educational objectives: The classification of educational goals: Handbook 1: Cognitive domain*. D. McKay, 1956.
[5] J. Sweller, P. Ayres, and S. Kalyuga, "Intrinsic and Extraneous Cognitive Load," *Springer New York*, 2011.
[6] R. Moreno, "Decreasing cognitive load for Novice Students: Effects of Explanatory versus Corrective Feedback in Discovery-Based Multimedia," *Instructional Science*, vol. 32, no. 1-2, 2004, pp. 99-113.
[7] L. Wang and J. Wang, "Teaching Reform of Embedded Operating Systems under Constructivism and MOOC Modes," *Comput--er Era*, no. 10, 2016, p. 4. (In Chinese)
[8] C. Liu, D. Zhang, D. Wu, et al., "Project-Based Teaching Reform and Practice of Mechanical Principles Course for Cultivating Innovation Abilities," *The Guide of Science & Education* , vol. 34, 2023, pp. 130-133. (In Chinese)
[9] Y. Zhu, "Application attempt of project-driven teaching model based on constructivism theory in Higher Vocational Computer Software Teaching," *Science & Technology Vision*, vol. 26, no. 2, 2016, pp. 2. (In Chinese)
[10] Y. Gao, X. Yin, L. Song, et al., "Exploration and Practice of Stress Medicine Course Construction Based on Constructivism Theory," *China Medical Herald*, vol. 21, no. 10, 2024, pp. 28-31. (In Chinese)
[11] Y. Chen, "Constructivist Learning Theory and Mental Health Education," *Theory and Practice of Education*, vol. 44, no. 2, pp. 123-126, 2024. (In Chinese)
[12] Y. Wei and Z. Peng, "Application of the flipped classroom model based on Bloom's Taxonomy of Educational Objectives in endodontics education for undergraduate dental students," *PeerJ*, vol. 13, 2025, e18843.
[13] S. Fang, "Practice of Rain Classroom Teaching Mode Based on Bloom's Cognitive Objectives-Taking 'Ancient Chinese Literature' as an Example," *Journal of Chizhou University*, vol. 38, no. 5, 2024, pp. 140-142. (In Chinese)
[14] M. Wang, "Research on Blended Teaching Design Based on Bloom's Taxonomy-Taking Information Technology Course as an Example," *Computer Knowledge and Technology*, vol. 19, no. 25, 2023, pp. 159-162. (In Chinese)
[15] L. Li and W. Cui, "Teaching Design Strategies of Undergraduate Courses Based on Cognitive Load Theory-Taking Biochemistr--y Course as an Example," *Chemistry of Life*, vol. 45, no. 2, 2025, pp. 371-380. (In Chinese)
[16] L. Yang, G. Si, C. Wang, P. Huang, K. Xiao, and Z. Du, "Application of Cognitive Load Theory in the Experimental Teaching Design and Application of Computer Organization Principles," *Research and Exploration in Laboratory*, vol. 44, no. 1, 2025, pp. 128-131. (In Chinese)
[17] W. Kong and Y. Wu, "Thirty Years of Foreign Teacher Feedback Research: Theory and Practice," *College English teaching and*

research, vol. 64, no. 3, 2013, pp. 90-96. (In Chinese)

[18] X. Huang and Z. Li, "Strategies for Building Effective Teaching: Feedback Theory," *Research in Educational Development*, vol. 2011, no. 4, 2011, pp. 43-47. (In Chinese)

[19] X. Jiang, M. Shao, and J. Shi, "'AI+Digital Twin' Empowering University Smart Education: Iterative Integration, Application Scenarios, and Practical Approaches," *Journal of Heilongjiang Institute of Technology*, vol. 39, no. 1, 2025, pp. 66-80. (In Chinese)

[20] J. Liu and C. Bian, "AI-Empowered Teaching Reform and Practice Exploration of Applied Undergraduate Computer Network Course," *Journal of Higher Education*, vol. 2024, no. 32, 2024, pp. 148-151. (In Chinese)

[21] C. Hu and Z. Zhou, "Research on the Path of AI-Empowered Innovation and Development of Higher Vocational Education-Based on the Application Perspective of ChatGPT," *Office Automation*, vol. 2024, no. 23, 2024, pp. 73-79. (In Chinese)

[22] Z. Zheng, "Application of AI Technology in the Field of Education under the Background of Big Data," *Information China*, vol. 2024, no. 12, 2024, pp. 190-192. (In Chinese)

[23] S. Wang, "Application of Artificial Intelligence Technology in Education," *High-Technology & Industrialization*, vol. 2024, no. 12, 2024, pp. 85-87. (In Chinese)

AUTHOR'S PROFILE

Lei Liu, earned his Ph.D. from the Department of Applied Physics at Eindhoven University of Technology (TU/e), the Netherlands, in 2013. He is currently a lecturer at China's School of Data Sciences, Zhejiang University of Finance and Economics. His research interests span data mining, natural language processing and deep learning.