Journal of Contemporary Educational Research, 2026, Volume 10, Issue 4
http://ojs.bbwpublisher.com/index.php/JCER
Online ISSN: 2208-8474
Print ISSN: 2208-8466

Teaching Innovation and Practice of Computer Network Course from the Perspective of Knowledge Visualization

Gongjin Chen*, Yantao He, Pengteng Huang

Department of Computing, Guangdong University of Science and Technology, Dongguan, China

***Corresponding author:** Gongjin Chen, chengongjin@gdust.edu.cn

Abstract: As the core basic course of computer science, the computer network course has long been faced with teaching dilemmas such as many and fragmented knowledge points, abstract and difficult to understand protocol levels, and a disconnection between theory and practice. Aiming at these problems, this study constructs a knowledge framework ontology covering all levels of TCP/IP protocol stack, designs a hierarchical and multidimensional visual teaching system, and carries out quasi-experimental research in university classrooms to verify its teaching effect. The research adopts the methodology combining the design research method and quasi-experiment method, and develops an interactive visualization platform based on knowledge graphs, which supports the dynamic demonstration of protocol flow, layer-by-layer analysis of packet structure, interactive exploration of network topology, and other functions. The experimental results show that the visual teaching group is significantly better than the traditional teaching group in the dimensions of agreement process understanding and cross-layer concept association, especially for learners with weak spatial ability. This study provides a reusable theoretical framework and practical paradigm for the knowledge visualization teaching of computer courses.

Keywords: Computer network; Knowledge visualization; Knowledge graph; Teaching application; Cognitive load; Quasi-experimental study

Online publication: May 19, 2026

1. Introduction

1.1. Research background and question proposal

As a basic required course for computer science and technology, network engineering, and other majors, the computer network course covers the seven-layer model of OSI and the four-layer protocol stack of TCP/IP, involving hundreds of protocols and concepts such as physical transmission, data link, network interconnection, end-to-end transmission, application services, etc. However, with the rapid development of Internet technology, the course content is constantly expanding. From the traditional wired network to wireless network, mobile Internet, Internet of Things, and other emerging fields, the density and complexity of knowledge continue to increase, but in the current teaching practice, there is a common dilemma of "difficult for teachers to speak, difficult for students to understand." The structural thinking of protocol level is difficult to establish, the dynamic process of data encapsulation and unencapsulation lacks intuitive perception, and the abstract mechanisms such as routing algorithm and congestion control are difficult to understand, which leads to the decline of students' interest in learning, the superficial knowledge grasp, and the weak engineering practice ability. Although the existing teaching reform introduces Wireshark packet capture analysis, Cisco simulator, and other tools, most of them are discrete skill training, lack a systematic knowledge framework support, and fail to effectively connect theoretical concepts and engineering practice. Knowledge visualization, as an important direction of educational technology, has achieved remarkable results in mathematics, physics, and other disciplines. However, the application in the field of computer networks is still in its infancy, and there are problems such as fragmentation of knowledge representation, disconnection between visualization and teaching strategies, and lack of empirical effect verification. Therefore, it is urgent to build a systematic computer network knowledge framework and design a visualization scheme that deeply integrates teaching scenarios to solve the teaching problems of this course [1].

1.2. Research purpose and significance

This research aims to build a set of knowledge framework ontology covering the whole course of computer networks, design a hierarchical, multidimensional, and interactive visualization teaching system, explore the deep integration mode of visualization technology and teaching strategy, and verify its teaching effect through empirical research. The specific objectives include the establishment of a clear hierarchy and a correlation knowledge graph model. A visualization platform supporting dynamic demonstration of protocols and exploration of knowledge structure was developed to form teaching models such as flipped classroom and inquiry learning, and to provide empirical data and methodological reference for visual teaching effect evaluation. Its theoretical significance lies in expanding the application theory of knowledge visualization in computer professional education, constructing the methodological framework of network course knowledge modeling, and enriching the empirical basis of multimedia learning cognitive theory in the field of engineering education. Its practical significance is to provide directly usable visual teaching resources and platform tools for computer network courses in colleges and universities. It provides a transferable design paradigm for the visual teaching reform of similar computer professional courses (such as operating systems, compiler principles), and provides demand analysis and function design reference for educational technology enterprises and universities to cooperate in the development of intelligent teaching products [2].

1.3. Research status at home and abroad

In terms of foreign research, MIT OpenCourseWare integrated network protocol animation into online courses earlier. Packet Tracer simulator developed by Cisco Networking Academy realized the visual construction of network topology and protocol simulation. The OLI project of Carnegie Mellon University adopts knowledge tracking technology to optimize the learning path. In recent years, Stanford, ETH Zurich, and other universities have widely applied D3.js and other tools in the MOOC platform to develop interactive visual components to support students in exploring the protocol interaction process independently. Some courses introduce Flash animation to demonstrate the process of triple handshake and routing convergence. Tsinghua University, Zhejiang University, and other universities have tried multi-screen interaction and real-time data visualization in the construction of smart classrooms. Some scholars have carried out knowledge sorting research based on concept maps and mind maps. However, the existing research mainly focuses on the animation demonstration or tool development of a single knowledge point, lacks a systematic knowledge framework design covering the whole content of the course, the research on the teaching application strategy of visual resources is weak, and there is little rigorous empirical research to verify its actual effect. The main shortcomings of the current research are that the knowledge organization is fragmented and the computable and reasoning knowledge graph is not formed, the visualization design emphasizes presentation and ignores interaction, and lacks the exploratory function to support deep learning, the teaching application and visualization resources are separated, and the closed loop of "design-implementation-evaluation" is not formed, and the effect verification relies on subjective evaluation and lacks control experiments and cognitive mechanism analysis. These gaps are the directions that this study tries to break through [3].

1.4. Research content, method, and technical route

The core content of this research includes four interrelated modules: one is the ontology modeling of the computer network knowledge framework to clarify the knowledge domain division, node attributes, and association relationships; The second is the design and development of a visual teaching system to realize the multidimensional visual representation and interactive exploration of knowledge. The third is the design and implementation of teaching application mode, which integrates visualization resources into flipped classrooms, inquiry learning, and other scenes. The fourth is the quasi-experimental evaluation of teaching effect to verify the influence mechanism of visualization on learning effect and cognitive process. The research method is a mixed research design. The theoretical framework is constructed through literature research and expert consultation in the early stage, the visualization prototype and teaching plan are iteratively developed by design research method in the middle stage, and the quasi-experimental research method is used to compare and analyze the teaching effect in the later stage. Supplemented by interviews and observation to collect qualitative data, the technical route follows the iterative cycle of "analysis-design-development-implementation-evaluation." Firstly, the characteristics of course knowledge and teaching requirements are analyzed, the knowledge ontology and visual representation strategy are established, and then the system architecture and interactive interface are designed, and the visual prototype is developed. Finally, the effect was evaluated and the feedback optimization was given to form a scalable result. The research tools included Neo4j graph database, D3.js visualization library, Python data analysis toolkit, and SPSS/AMOS statistical software [4].

2. Theoretical basis and analysis framework

2.1. Definition of core concepts

Knowledge framework refers to a systematic knowledge system organized according to the internal logical structure around a specific subject field. It goes beyond the scattered factual knowledge and reveals the hierarchical relationship, causal association, and dynamic evolution between concepts. In this study, the knowledge framework specifically refers to the knowledge ontology of computer network courses, including protocol hierarchy, core concept definitions, knowledge point attributes, and their multidimensional associations. Educational visualization is the process of transforming abstract teaching content into visual symbols, graphical representations, or dynamic images. The essence of educational visualization is to enhance the perceptibility and understandability of information through visual channels, reduce cognitive load, and promote deep processing. Cognitive Load Theory distinguishes intrinsic load (the complexity of the task itself), extrinsic load (the redundant processing caused by poor instructional design), and relevant load (the promotion of meaningful processing of schema construction), which provides a theoretical basis for optimizing the cognitive efficiency of learning resources for visual design [5].

2.2. Theoretical basis

Constructivism learning theory emphasizes that knowledge is a meaning system constructed by learners through active exploration and collaborative communication in a specific situation, which requires that visual design should provide an interactive environment that supports students to independently discover concept associations, verify hypotheses, and reflect on understanding rather than one-way knowledge presentation. Dual coding theory points out that there are two independent and closely related processing channels of speech and non-speech in the human cognitive system, and activating both channels at the same time can form redundant coding representation to significantly improve the effect of information retention and extraction, which provides a physiological basis for the design of multimedia teaching resources. The cognitive theory system of multimedia learning proposed 12 design principles including multimedia principle, spatial proximity principle, temporal proximity principle, and consistency principle, which provided operational guidelines for the design of the visual interface in this study. The SOI model further explains the cognitive process of deep learning, that is, learners first select relevant information, organize it into coherent mental representations, and finally integrate it with prior knowledge. Visualization tools can effectively support this information processing process by guiding attention, visualizing structure, and providing anchors [6].

2.3. Analysis of knowledge characteristics of Computer Network Course

The knowledge system of computer network course has distinct hierarchical characteristics. The vertical layering of TCP/IP protocol stack from physical layer to application layer constitutes the backbone structure of the course. Each layer provides services to the upper layer and shields the implementation details of the lower layer. The three elements of protocol—syntax, semantics, and timing—constitute the core knowledge type of the course, which requires students to master static structure description and dynamic process deduction at the same time. The systematization of knowledge is embodied in the consistent application of cross-layer concepts such as end-to-end principle, hierarchical thinking, encapsulation and de-encapsulation, and the dynamicity is embodied in the life cycle processes such as connection establishment, data transmission, error recovery, and connection release. Based on the above characteristics, the important and difficult points in teaching can be summarized into three categories: the concretization of abstract concepts, the explication of implicit processes, and the simplified modeling of complex systems. The visualization design needs to expand the time dimension into a spatial sequence, map the logical structure into a visual form, and aggregate the distributed process into a traceable animation demonstration [7].

2.4. Visual teaching analysis framework

This study constructs a four-dimensional analysis framework of “knowledge dimension, representation dimension, interaction dimension, and teaching dimension” as an integrated tool connecting theory, design, and practice. The knowledge dimension focuses on the integrity and accuracy of ontology modeling to ensure that the visualization content covers the requirements of the curriculum outline and the concept definition is authoritative. The representation dimension focuses on the effectiveness of visual encodings to select appropriate graphical grammars to map different types of knowledge structures. The interaction dimension focuses on the depth of learners' participation to design multi-level interaction from overview to detail, from observation to operation, and from individual to collaboration. The teaching dimension focuses on the appropriateness of application scenarios to embed visualization resources into the complete teaching process of preview, teaching, review, experiment, evaluation, and so on. The framework emphasizes the collaborative optimization of four dimensions, that is, the granularity of the knowledge dimension determines the complexity of the representation dimension, the form of the representation dimension restricts the possibility of the interaction dimension, the function of the interaction dimension serves the goal of the teaching dimension, and the feedback of the teaching dimension drives the iterative refinement of the knowledge dimension, which overcomes the limitation of the existing research that considers visualization as a simple "teaching aid tool." Reposition it as a core medium for reshaping the cognitive process of teaching [8].

3. Knowledge framework construction and visualization design

3.1. Principles of knowledge framework design

The design of the knowledge framework follows four basic principles. The systematic principle requires covering all layers of the TCP/IP stack, from the physical transmission medium to the application layer protocol, from the traditional Internet to the expansion of emerging technologies, so as to form an unbroken knowledge coverage. The hierarchical principle emphasizes the three-level organization of macro architecture, meso modules, and micro knowledge points: the macro level shows the overall structure of the protocol stack and the relationship between layers, the meso level shows the internal core mechanisms and algorithms of each layer, and the micro level is refined to the data format, field meaning, and state transition of specific protocols. The relevance principle aims to reveal the vertical inter-layer service interface relationship, the horizontal comparison relationship of the same layer protocol, the procedural data life cycle relationship, and the application scene mapping relationship, and break the knowledge island. The practical principles ensure that each knowledge point is aligned with real network scenarios and experiments, such as linking HTTP protocols to browser developer tools and routing protocols to GNS3 simulator configurations.

3.2. Modeling of knowledge ontology

The knowledge ontology adopts a four-level structure of "domain-topic-concept-attribute." The seven domains of knowledge include: Network infrastructure (architecture, performance metrics, topology type), physical layer (transmission medium, coding modulation, multiplexing), data link layer (framing, error control, flow control, MAC protocol), network layer (IP addressing, routing algorithm, ICMP, NAT, IPv6), transport layer (UDP/TCP protocol, reliable transport, congestion control), application layer (DNS, HTTP, FTP, SMTP, P2P, Socket programming), and network security (encryption system, digital signature, firewall, intrusion detection). Each knowledge node defines five types of attributes: concept definition (authoritative textbook expression), mathematical model (formula and algorithm pseudo-code), protocol format (group header structure diagram), application scenario (typical use cases and configuration examples), and common problems (students' easy-to-confuse points and typical errors).

Four kinds of relationships are constructed in knowledge association network. Vertical association is the indirect interface through the "service access point" concept explicit layer, such as the process of IP datagrams encapsulated into Ethernet frames. Horizontal correlation to compare the differences between the same layer protocols, such as TCP and UDP in reliability, connectivity, and flow control. Procedural associations trace the end-to-end journey of a packet, such as the complete chain of DNS resolution, TCP connection, HTTP request, IP routing, and MAC forwarding during a Web access. Apply correlation to link real network tools with phenomena, such as Wireshark filter expressions corresponding to protocol fields. These associations are represented by labeled directed edges in the knowledge graph, which supports graph traversal queries and path recommendations.

3.3. Visual design strategy

The hierarchical mapping strategy transforms the protocol hierarchy into a visual hierarchy: vertically, a vertically stacked layout is adopted, with the upper layer below (close to the user) and the lower layer above (close to the physical medium), which conforms to the cognitive metaphor of "the bottom supporting the top." Layers are color-coded (e.g., physical layer-red, link layer-orange, network layer-yellow, transport layer-green, and application layer-blue) to create a stable visual schema. For example, SYN-SYN/ACK-ACK interaction of TCP three-way handshake is presented as a message passing animation between nodes, and the control of pause, rollback, and slow playback is supported with the timeline sliders. The interactive exploration strategy provides multi-scale navigation. The overview mode displays the global knowledge graph and supports zooming and panning. Focusing mode expands the detailed information and association network of a single knowledge node. Contrast mode juxtaposition shows the difference between related concepts such as distance vectors and link-state routing algorithms. Multi-view collaborative strategy integrates protocol stack hierarchy graph, network topology graph, time sequence interaction graph, and data package structure graph in the same interface, and realizes state synchronization by brushing and linking technology.

3.4. Design of visual prototype system

The system was developed with a front-end and back-end separation architecture. In the data layer, the Neo4j graph database was used to store knowledge ontology, the node types covered knowledge concepts, protocol entities, algorithm processes, and application scenarios, and the relationship types included inclusion, dependency, comparison, and application relationships. The logical layer encapsulates RESTful APIs such as knowledge retrieval, path planning, and user behavior recording based on the Python Flask framework, and integrates the NetworkX library to realize the graph algorithm analysis required for personalized recommendations. The presentation layer uses React and D3.js technology stack to ensure the interactive fluency of a large-scale knowledge graph through Canvas and SVG collaborative rendering. The core components of the system include a vertically stacked protocol stack hierarchy diagram, a force-directed network topology diagram that can simulate link failures, a sequential interaction diagram that supports stepwise execution, a packet structure diagram that can parse the meaning of fields layer by layer, and a conceptual relationship diagram that maps the importance of nodes. Dynamic demonstration cases are designed for typical scenarios such as TCP connection establishment, IP datagram fragmentation, and routing algorithm convergence.

3.5. Technical implementation scheme

Technology selection takes into account both development efficiency and operational performance. Neo4j Community Edition was used as the graph database, and its native graph storage and index mechanism was used to efficiently process complex association queries. React 18 and TypeScript were used as the front-end framework to ensure component development and type safety. The visualization engine used D3.js v7 to handle custom graphics rendering, AntV G6 to handle large-scale network topology, and ECharts to handle statistical charts. The back-end service was deployed in Docker containers, and Nginx was used to achieve reverse proxy and load balancing. The key function implementation includes: the visual rendering of the knowledge graph adopts the hierarchical layout algorithm, which is divided into layers according to the protocol level, and then the node position is forced to optimize in the layer. Learning path recommendation is based on Knowledge Tracing model, which uses an LSTM network to predict students' mastery probability of each knowledge point and generate personalized review paths. The real-time collaboration function uses WebSocket protocol to realize multi-user synchronous editing and cursor sharing. The system interface uses a dark theme to reduce visual fatigue, key elements use high-contrast colors to highlight, and interactive feedback is controlled within 200 ms to ensure a smooth experience.

4. Teaching application design and implementation

4.1. Teaching application scenarios and modes

The visualization system forms a closed-loop support throughout the whole process of teaching. In the preview stage before class, students conduct a global overview through knowledge graph navigation, the system highlights the core nodes to be learned according to the teaching progress, recommends the associated prerequisite knowledge for review, and students can mark doubts for classroom focus. In the teaching stage, the teacher used dynamic visualization to replace the traditional blackboard writing, such as adjusting the network load parameters in real time when explaining TCP congestion control, observing the changes of the congestion window curve, and presenting the algorithm pseudo-code and graphical representation synchronously. Students can autonomously control the playback speed by scanning the code or synchronously viewing the webpage. In the review stage after class, the system generated personalized knowledge mastery portraits based on classroom interaction data and recommended concept comparison views of weak links and exercise drills. Students can initiate a "packet journey" exploration mission, tracing from application layer request to physical layer signal transmission, and consolidating cross-layer understanding. In the experimental teaching stage, the visualization platform was deeply integrated with Wireshark, GNS3, and other tools. The real data packets captured by Wireshark could be imported into the system for analysis, and the structure of each layer of the protocol stack was automatically matched. GNS3 configuration of the network topology can be synchronized to the visual interface, real-time display routing table changes, and packet forwarding path.

4.2. Innovation of teaching mode

Flipped classroom model reconstructs the time sequence of knowledge transfer and internalization. Before class, students watched 15–20 minutes of micro-lecture videos (embedded with interactive visualization components), completed a concept detection quiz, and the system automatically analyzed error patterns to adjust classroom focus. In the class, the teacher no longer repeated the basic explanation, but organized the exploration activities based on the visual environment, such as the group task of "design a reliable transmission protocol," students dragged components on the visual platform, simulated operation, and observed the performance. After class, students improve their personal knowledge map notes and share them with the learning community. Inquiry learning is driven by real questions, such as "Why is the video call slow but the download speed is normal?" Students use visualization tools to track the different behaviors of UDP and TCP streams, measure delay, jitter, packet loss rate, and discover how QoS mechanisms work. Collaborative learning supports multiple people to simultaneously edit and share the knowledge map in the class. Students are responsible for the improvement and annotation of different knowledge domains, and teachers can view the editing history and contribution to form a collective wisdom precipitation.

4.3. Integration and deployment of teaching resources

The platform provides a standardized content encapsulation format (education data specification based on JSON-LD), supports LTI standard docking with mainstream MOOC platforms (Xuetangonline, Wisdom Tree, Superstar), and embeds visualization components into course pages in the form of iframe or Web Component. For mobile terminal learning scenarios, lightweight H5 applications are developed to retain core knowledge graph browsing and simple interaction functions to adapt to fragmented learning needs. The resource update mechanism uses Git version control, the teacher community can contribute new visualization cases and merge the main branch after expert review, forming a continuous evolution of resource ecology.

4.4. Teaching implementation cases

The “transport layer” unit is taken as an example to show the complete implementation process. Before class, students previewed the protocol format comparison diagram of TCP and UDP through the visualization system and completed the three-way handshake sequence sorting exercise. The system records the common errors of students misclassifying “four wave” as “three wave.” In the class, the teacher focused on the connection release mechanism based on the preview data, and used the time sequence interaction diagram to demonstrate the role of FIN flag interaction and TIME_WAIT state. Students discussed “why 2MSL waiting time is needed” in groups, and observed the packet loss caused by premature closure by adjusting the visual parameters. In the experiment, students build a topology in GNS3 to generate TCP flows, and import the captured packet results into the system for analysis to verify the dynamic changes of the sliding window and confirm the accumulation mechanism. After class, the system recommends a comparative learning task of congestion control algorithms, and students explore the transition conditions of four states: slow start, congestion avoidance, fast retransmission, and fast recovery, and draw state machine diagrams. Unit tests show that the experimental class has a 23% increase in the scoring rate of application questions such as header field calculation and state transition analysis compared with the control class. The model architecture case diagram is shown in **Figure 1**.

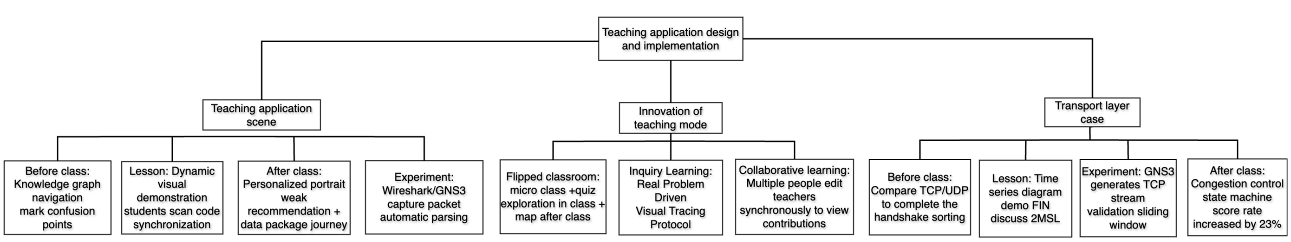

Figure 1. Example diagram of the model architecture

5. Evaluation of teaching effect

5.1. Evaluation index system

The evaluation uses the framework of Kirkpatrick’s four-level model combined with cognitive load theory.

In the reaction layer, students' satisfaction and willingness to use the visualization system were measured by questionnaires. In the learning layer, the knowledge mastery was evaluated by comparing the pre-test and post-test scores, and the test questions were divided into three levels: memorability (protocol port number), understanding (explaining the principle of slow start), and application (calculating the congestion window). The behavior layer evaluated the protocol analysis ability and troubleshooting ability through classroom observation and experimental report. The results layer tracks final exam scores and subsequent professional course performance. Cognitive load was measured using subjective scales (simplified version of NASA-TLX) to assess mental effort, combined with pupil diameter changes from eye movement data (if available) and fixation heatmaps to analyze attention allocation. The deep learning indicators include the quality of concept map drawing (the number of nodes, the number of connections, and the rationality of hierarchical structure) and the performance of migration problem solving (the design of new transport protocols).

5.2. Research design

A non-randomly assigned control group pre-test and post-test quasi-experimental design was used. Two parallel classes were selected in the College of Computer Science in a university: the experimental class (n = 62) used a visual teaching system, and the control class (n = 58) used traditional PPT+ blackboard teaching. The two classes were taught by the same teacher, and the textbooks, class hours, and assignments were the same. The pre-test included the computer network knowledge test and the Spatial ability Test (VZ-2), and confirmed that there was no significant difference between the two groups. The experiment lasted 16 weeks and covered the full semester course. Intervention program: the experimental class used knowledge graph navigation before class preview, teachers used visual demonstration in class, and embedded interactive exploration tasks in homework after class. The control class was carried out according to the conventional teaching process. The post-test included a standardized knowledge test, concept map drawing, protocol analysis practical assessment, and learning motivation and cognitive load questionnaire.

5.3. Data collection and analysis

Quantitative data collection: pre-test scores, test scores of each unit, system use log (access frequency, stay time, interaction path), eye movement experiment data (subsample n = 20). Qualitative data collection: semi-structured interviews (focus groups of six people per class), analysis of learning logs, and coding of classroom videos. Analysis of covariance (ANCOVA) was used to control the influence of pretest scores and spatial ability. Repeated measures analysis of variance was used to test the time effect of learning gain. Structural equation model (SEM) was used to test the mediation path of "visual use → cognitive load → learning achievement." Thematic analysis was used for qualitative data, and the coding categories included visualization use strategies, perceived usefulness, difficulties encountered, and suggestions.

5.4. Discussion of results

The quantitative results showed that the total score of the experimental group was significantly higher than that of the control group (Cohen's d = 0.68, medium effect size), especially in the application topics (protocol design, fault diagnosis). The analysis of cognitive load showed that the external load score of the experimental group was significantly lower than that of the control group, and the related load score was significantly higher than that of the control group, indicating that the visualization effectively reduced invalid cognitive processing and promoted deep thinking. The analysis of the moderating effect of spatial ability showed that the improvement of students with low spatial ability (Δ = 18 points) was significantly greater than that of students with high spatial ability (Δ = 7 points), indicating that visualization had a compensatory effect on learners with insufficient cognitive resources. Qualitative analysis found that students highly recognized the value of dynamic presentation for understanding the abstract process, but also pointed out that some animations were too fast and lacked pause prompts. Some students rely too much on visualization and ignore the close reading of the protocol text, suggesting the need to design a guided scaffold. The comprehensive results show that the effectiveness of visualization teaching depends on the quality of design and the depth of teaching integration, rather than the simple application of technology itself.

6. Conclusion

Aiming at the teaching dilemma of computer network course, this study constructs a systematic knowledge graph and visualization teaching platform, realizes the multidimensional visual representation of protocol level, data flow, and network topology, and confirms the significant effect of visualization in improving conceptual understanding and reducing external cognitive load through teaching experiments. The follow-up research will explore the personalized tutoring mechanism driven by artificial intelligence, develop the virtual reality network experiment environment, expand the knowledge framework to cover the emerging technology fields, promote the open source sharing and continuous evolution of visual teaching resources, and finally form a new form of intelligent and immersive computer network teaching.

Funding

This work was supported by the Quality Engineering Project of Guangdong University of Science and Technology (GKZLGC2025025, GKZLGC2025028, and GKZLGC2025043).

Disclosure statement

The authors declare no conflict of interest.

References

[1] Zhou Z, Feng D, Liu Y, 2025, Reconstruction and Practice of Interdisciplinary Course Content System Framework Based on Engineering Logic——Taking the Course of Opto-Electronic Instrument System Design as an Example. University Education, (22): 39–42 + 48.

[2] Li J, Cheng B, Liu L, 2020, Knowledge Graph Empowering the Construction of General Education Courses in Universities——Taking the Course of "Social Organization Management" as an Example. Coal Higher Education, 43(04): 63–71.

[3] Li HX, 2020, A new Perspective on Improving the Effect of Ideological and Political Teaching in Courses——Based on the Framework of Students' Knowledge Absorption Ability. Journal of Tianjin Chengjian University, 31(03): 231–236.

[4] Li Y, He X, Fang J, 2025, Theoretical Framework and Practical Application of Generative Artificial Intelligence Empowering the Construction of Smart Courses in Colleges and Universities——Taking the C Language

Programming Course as an Example. Teacher Education Forum, 38(03): 26–33.

[5] Song H, Sun H, Xu J, 2024, Research on Curriculum Ideological and Political Construction Based on Curriculum Map——Taking the Course of “Aircraft Structural Mechanics” as an Example, Proceedings of the Fifth National Forum on Ideological and Political Teaching Reform of Aerospace Courses (II), School of Aerospace and Civil Engineering, Harbin Engineering University, 182–186.

[6] Wang F, 2020, Research on Entrepreneurship Curriculum System in Applied Undergraduate Colleges——Based on the Theoretical Framework of Knowledge Management. Vocational Education Forum, 36(12): 58–66.

[7] Yang M, Du G, Zhang A, et al., 2020, Exploration and Practice of Teaching Methods of Computer Network Course. Heilongjiang Science, 11(15): 78–79.

[8] Wang J, 2023, Research on the Construction of Course Knowledge Network Based on Knowledge Visualization, dissertation, Inner Mongolia Normal University.