ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0049

# Construction and Empirical Research on the Interactive Influencing Factors Model of Programming Teaching Based on Intelligent Conversation

Bicheng Zhuo

Graduate student, School of Education, China West Normal University; 637000, China

## Abstract

**This study first constructs a hypothesis model examining interactive factors in intelligent conversation-based programming instruction. Through questionnaire surveys and structural equation modeling, we validate the model's hypotheses. Results demonstrate that instructional interactions are primarily shaped by learning motivation, programming self-efficacy, and teacher factors, while intelligent technologies, teacher influence, and learning environments significantly impact students' motivation and programming self-efficacy. Finally, we propose recommendations for optimizing instructional interactions through intelligent technologies, teacher engagement, and environmental design to enhance the effectiveness of intelligent conversation-based programming education.**

## Keywords

**Intelligent conversation; Programming teaching; Teaching interaction; Influencing factors.**

## 1. Introduction

Artificial intelligence has emerged as a pivotal force driving the new wave of technological revolution and industrial transformation. As a critical tool bridging humans and machines to achieve intelligent capabilities, programming's significance continues to grow. The State Council's "New Generation [1] Artificial Intelligence Development Plan" proposes implementing a nationwide intelligent education initiative, introducing AI-related courses in primary and secondary schools while gradually expanding programming education. Generative AI-powered conversational systems enable natural language input, interaction, and generation to assist coding and teaching, providing more efficient learning experiences for both students and educators. Consequently, guiding students to recognize, understand, and apply intelligent conversational systems for collaborative innovation has become a key strategy in programming education. This study investigates the internal relationships and mutual influences among teaching interaction factors in AI-driven programming classrooms through model construction and empirical validation, aiming to develop effective teaching strategies that enhance programming education outcomes.

## 2. Research Hypothesis and Model Determination

### 2.1. Determination of the Hypothesis of the Interaction between the Factors and Teaching

Teaching interaction refers to the dynamic exchanges between students and their learning environment, encompassing interactions among students and [2] teachers, peer-to-peer communication, as well as engagement with materialized resources. The influencing factors of teaching interaction operate across multiple dimensions. This study categorizes the factors

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0049

affecting programming teaching interaction based on intelligent conversation into two main types: external and internal factors. External factors include intelligent technologies, teacher influence, and environmental support, while internal factors involve learning motivation and programming self-efficacy.

### 2.1.1. Intelligent Technology

Technical factors have long been pivotal in classroom teaching [3] interactions. Gu Xiaoqing et al. demonstrated that intelligent tools can effectively stimulate students' learning interest and motivation, enhancing their engagement at emotional, cognitive, and behavioral levels. Based on this, the study proposes the following hypotheses: (H1) Intelligent technology significantly enhances programming learning motivation during interactive processes; (H2) It significantly boosts programming self-efficacy; and (H3) It significantly improves teaching interactions.

### 2.1.2. The Influence of Teachers

Teachers play a pivotal role in the teaching process, where their pedagogical philosophy, instructional methods, and classroom management skills profoundly influence instructional interactions. By providing support and guidance, teachers can help students better adapt to intelligent learning environments and enhance learning outcomes. Numerous studies have demonstrated a significant positive correlation between teacher support and students' overall motivation. Based on this, the present study proposes the following hypotheses: (H4) Teacher factors in programming learning interactions significantly positively influence learning motivation; (H5) Teacher factors significantly positively influence programming self-efficacy; (H6) Teacher factors significantly positively influence instructional interactions.

### 2.1.3. Environmental Support

The classroom serves as a pivotal space for learners to acquire knowledge, develop skills, and achieve holistic growth. Key environmental elements—including physical infrastructure and cultural ambiance—should be designed and optimized to facilitate teaching interactions while supporting learners 'academic progress and well-being. Establishing a harmonious learning environment is fundamental to nurturing students' motivation. Based on this premise, the study proposes the following hypotheses: (H7) Environmental support during programming learning interactions significantly enhances learning motivation; (H8) It significantly boosts programming self-efficacy; and (H9) It significantly improves teaching interactions.

### 2.1.4. Learning Motivation

A learner's motivation directly influences their engagement in teaching interactions, affecting both initiative and participation levels, which ultimately impacts learning [4] outcomes. Ambiguous motivation directly hinders the improvement of teaching interaction quality. Based on this, the study proposes the following hypothesis: Motivation has a significant positive effect on teaching interactions during programming learning (H10).

### 2.1.5. Programming Self-efficacy

Students with high self-efficacy believe they can complete programming tasks. Even when facing complex programming challenges, they maintain a positive attitude, actively participate in interactive activities, and demonstrate strong learning engagement, leading to better programming outcomes. Moreover, self-efficacy is closely linked to learning motivation, and enhancing both motivation and self-efficacy forms the foundation for effective programming education in classrooms. Based on this, the study proposes the following hypotheses: (H11) Programming self-efficacy significantly positively influences learning motivation during programming learning interactions; (H12) Programming self-efficacy significantly positively impacts teaching interactions.

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0049

## 2.2. Determination of the Hypothesis Model

In the 1990s, Moore M.G. first categorized instructional interactions into three types: student-teacher, student-student, and student-content interactions [5]. In recent years, with the application of new technologies like artificial intelligence and big data in education, the role of technological elements in instructional interactions has become increasingly prominent. Scholars such as Sun Tianlinzi introduced new-generation information technology elements, dividing online classroom interactions into human-computer interaction, interpersonal interaction, and human-cognition interaction [6]. Based on this, this study categorizes programming instructional interactions based on intelligent conversation into three types: interpersonal interaction, human-computer interaction, and human-cognition interaction. Integrating the above analysis, this study establishes a hypothetical model of influencing factors for programming instructional interactions based on intelligent conversation, as shown in Figure 1.

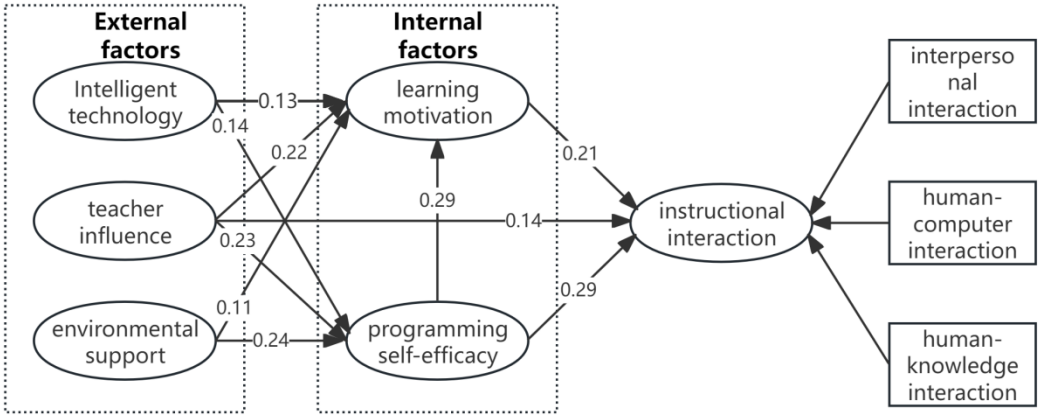


**Figure 1.** Hypothesized model of interactive factors in programming teaching based on intelligent conversation

# 3. RESEARCH PROCESS AND RESULTS

## 3.1. Research Tools and Survey Implementation

### 3.1.1. Research Methods and Tools

This study employs literature review and questionnaire survey methods. First, through literature analysis, we identify five dimensions of instructional interaction and construct a hypothesis model. Subsequently, we develop a formal [7] questionnaire by designing, distributing, and validating the data. Correlation and path analyses are then conducted to verify inter-factor relationships, culminating in research conclusions. The questionnaire, referencing the work of Xie Changjiu et al., is titled "Factors Influencing Instructional Interaction in Intelligent Conversation-Based Programming." It comprises three sections: student information, instructional interaction factors, and instructional interaction quality, with the latter two sections using a 5-point Likert scale.

### 3.1.2. Questionnaire Test

Following the completion of the questionnaire, a pilot test was conducted, yielding 76 valid responses. The analysis revealed that all factor scales demonstrated reliability coefficients exceeding 0.75, confirming their strong internal consistency. Items with a corrected item-total correlation (CITC) below 0.4 were removed. Subsequently, exploratory factor analysis was performed to assess structural validity, yielding KMO values of 0.734 for the Influencing Factors Scale and 7.23 for the Teaching Interaction Scale, both with significance levels of 0.00. After rotation of the initial factor loading matrix, the cumulative variance explained by factors

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0049

reached 67.330% and 62.915%, respectively. These results align with expectations, demonstrating robust reliability and validity.

#### 3.1.3. The Questionnaire Has Been Officially Distributed

The official survey adopted a combination of online and offline methods, with all respondents being junior high school students who had taken programming courses. A total of 357 questionnaires were collected, and after rigorous preprocessing, 14 invalid questionnaires were excluded, leaving 340 valid questionnaires, resulting in a validity rate of 96.04%.

### 3.2. Data Statistics and Analysis

#### 3.2.1. Validity and Reliability Analysis

The Cronbach's alpha coefficients for the Influencing Factors section and Teaching Interaction section were 0.797 and 0.899 respectively, indicating strong reliability for both the Influencing Factors Scale and the Teaching Interaction Quality Perception Scale. Confirmatory factor analysis yielded a KMO value of 0.921 with a significance level of 0.000. Through calculations of Average Variance Extraction (AVE) and Composite Reliability (CR), all dimensions demonstrated AVE values above 0.5 and CR values exceeding 0.7, confirming strong convergent validity. The Fornell-Larcker criterion test revealed that correlation coefficients between dimensions remained within acceptable thresholds, with item correlations consistently below the square root of AVE values. These results collectively demonstrate robust discriminant validity among questionnaire items.

#### 3.2.2. Descriptive Statistics

The study primarily targeted first and second-year junior high school students, with a gender ratio of 55:45. Nearly half of the participants had been exposed to programming within the past year. To determine whether gender, grade level, and programming duration were associated with various teaching interaction factors, independent samples t-tests were conducted for grade and gender. Results showed no significant differences across dimensions between students of different grades or genders. A one-way ANOVA of programming duration revealed a statistically significant ($p=0.01$) correlation between programming exposure duration and environmental variables, indicating that as students' programming time increased, their perceived and actual needs for programming environments also grew.

#### 3.2.3. Correlation Analysis

The Pearson correlation analysis assessed the relationships and their magnitudes between teaching interactions and various influencing factors. The results showed that teacher influence had a correlation coefficient of 0.462 with smart technology, while programming efficacy had a coefficient of 0.433 with teacher influence. Learning motivation showed a moderate correlation of 0.401 with environmental support, and programming efficacy demonstrated a moderate correlation of 0.491 with learning motivation.

**Table 1.** Correlation analysis results

| | intellectual technology | Teacher Influence | environmental support | academic motivation | Programming efficacy | human-computer interaction | interpersonal interaction | human-computer interaction |
|---|---|---|---|---|---|---|---|---|
| intellectual technology | 1 | | | | | | | |
| Teacher Influence | 0.462** | 1 | | | | | | |
| environmental support | 0.334** | 0.376** | 1 | | | | | |
| academic motivation | 0.321** | 0.353** | 0.401** | 1 | | | | |
| Programming efficacy | 0.368** | 0.433** | 0.390** | 0.491** | 1 | | | |
| human-computer interaction | 0.187** | 0.229** | 0.261** | 0.365** | 0.376** | 1 | | |
| interpersonal interaction | 0.151** | 0.195** | 0.176** | 0.342** | 0.284** | 0.428** | 1 | |
| human-computer interaction | 0.205** | 0.238** | 0.166** | 0.247** | 0.257** | 0.472** | 0.516** | 1 |

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0049

### 3.2.4. Model Evaluation and Hypothesis Testing

This study employed multiple fit indices including Chi-square Freedom Ratio (CMIN/DF), Root Mean Square Error (RMSEA), CFI, and GFI to evaluate the model's fit and explanatory power. As shown in Table 2, all metrics fell within the acceptable range, demonstrating that the structural equation model achieved satisfactory overall fit and demonstrated strong alignment with the actual data.

**Table 2.** Model fit test

| Metric | CMIN/DF | RMSEA | CFI | CFI | NFI | AGFI | RMR |
|---|---|---|---|---|---|---|---|
| Reference standard | <3 | <0.08 | >0.9 | >0.9 | >0.9 | >0.9 | >0.05 |
| Model index value | 1.127 | 0.019 | 0.99 | 0.907 | 0.924 | 0.893 | 0.052 |

This study employed structural equation modeling to test hypotheses, with results presented in Table 3. Overall, the P-values for paths H3 and H9 were both greater than 0.05, while all other paths showed P-values below 0.05. This indicates that neither intelligent technology nor environmental support significantly influenced instructional interaction, with all other hypotheses being valid. The findings demonstrate that teacher factors, learning motivation, and programming efficacy have direct and significant positive effects on instructional interaction. Additionally, intelligent technology, teacher influence, and environmental support all exert significant positive impacts on learning motivation and programming efficacy, while programming efficacy also significantly affects learning motivation.

**Table 3.** Path Coefficients and Hypothesis Testing

| way | | | Estimate | S.E. | C.R. | P | corresponding hypothesis | Established or not |
|---|---|---|---|---|---|---|---|---|
| Programming efficacy | <--- | intellectual technology | 0.143 | 0.054 | 2.636 | 0.008 | H2 | yes |
| Programming efficacy | <--- | Teacher Influence | 0.232 | 0.055 | 4.218 | *** | H5 | yes |
| Programming efficacy | <--- | environmental support | 0.242 | 0.055 | 4.385 | *** | H8 | yes |
| academic motivation | <--- | intellectual technology | 0.127 | 0.047 | 2.711 | 0.007 | H1 | yes |
| academic motivation | <--- | Teacher Influence | 0.216 | 0.049 | 4.4 | *** | H4 | yes |
| academic motivation | <--- | environmental support | 0.109 | 0.048 | 2.253 | 0.024 | H7 | yes |
| academic motivation | <--- | Programming efficacy | 0.289 | 0.057 | 5.063 | *** | H11 | yes |
| Instructional Interaction | <--- | academic motivation | 0.193 | 0.086 | 2.258 | 0.024 | H10 | yes |
| Instructional Interaction | <--- | Programming efficacy | 0.279 | 0.077 | 3.61 | *** | H12 | yes |
| Instructional Interaction | <--- | Teacher Influence | 0.132 | 0.064 | 2.075 | 0.038 | H6 | yes |
| Instructional Interaction | <--- | intellectual technology | 0.032 | 0.059 | 0.533 | 0.594 | H3 | deny |
| Instructional Interaction | <--- | environmental support | 0.007 | 0.061 | 0.121 | 0.904 | H9 | deny |

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0049

# 4. RESEARCH CONCLUSIONS AND RECOMMENDATIONS

## 4.1. Research Conclusion

### 4.1.1. Students' Requirements for Programming Environments Increase with Learning Duration

Descriptive statistics reveal no significant gender or age differences among junior high school students across all variables. However, as students gain programming experience, their perception of environmental factors increases significantly. This demonstrates that with growing coding proficiency, students' needs for stable environments, functional richness, and supportive atmospheres show a marked upward trend. A stable programming environment reduces technical glitches and distractions, allowing students to focus on coding and debugging. Additionally, abundant learning tools like intelligent chatbots and user-friendly code editors help students tackle complex programming tasks more efficiently. Finally, the supportive atmosphere plays a crucial role in programming education. In such environments, students can more easily access help, share experiences, and better navigate learning challenges.

### 4.1.2. The Teaching Interaction is Influenced By The Teacher Factor, Learning Motivation and Programming Efficacy

Teachers 'professional competence, instructional skills, classroom charisma, and support for students directly influence the quality of teaching interactions. These factors ultimately shape educational engagement by affecting students' learning motivation and programming self-efficacy. Students with stronger learning motivation demonstrate higher engagement in programming classes, investing more effort and enthusiasm into the curriculum, thereby enhancing the quality and effectiveness of teaching interactions. Moreover, students with strong self-efficacy can confidently tackle coding challenges and errors, skillfully apply programming knowledge to solve problems, and actively participate in classroom activities with greater initiative.

### 4.1.3. Smart Technology, Teacher Factors and Learning Environment Significantly Affect Learning Motivation and Programming Efficacy

This demonstrates that students 'learning motivation and programming efficacy are not solely influenced by teachers. The real-time cognitive support from smart technologies and resource-rich learning environments can significantly enhance students' psychological engagement. AI-powered conversations present code examples and execution results, helping students quickly grasp programming logic while promptly identifying syntax errors and guiding corrections. Consequently, these intelligent interactions effectively stimulate learning motivation and boost programming efficacy. When teachers acknowledge students 'perspectives and validate their contributions, it strengthens learning confidence, promotes self-directed learning, and fuels sustained motivation to achieve goals. Environmental factors encompass both hardware infrastructure and cultural contexts. Well-developed programming resources and equipment enhance students' confidence and drive to solve coding challenges independently. Moreover, a positive classroom atmosphere fosters peer interaction and increases participation in discussions. Being part of a trusting and cohesive community further boosts students' confidence in learning.

## 4.2. Advocacy

### 4.2.1. Perfect the "Human-machine Collaboration" of Precise Support and Intervention Guidance

Integrating intelligent conversational systems as a supplementary teaching tool promises to deliver real-time, personalized cognitive support for learners, thereby enhancing programming education outcomes. This approach requires educators to carefully select context-appropriate

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0049

conversational tools aligned with instructional objectives, while providing clear guidance on their optimal usage timing, methods, and strategies to prevent blind reliance on technology. Teachers should cultivate students' ability to ask high-quality questions through classroom guidelines and demonstrations. Students should be encouraged to avoid direct queries for answers, instead exploring indirect clues like "algorithmic logic," "potential error causes," and "optimization directions" to guide conversational systems toward strategic hints rather than standard answers. Educators must also guide students to critically evaluate conversational outputs—such as comparing generated code with student-written code in terms of efficiency, readability, and logical coherence—to prevent path dependence and maintain a healthy balance between technological assistance and cognitive autonomy.

#### 4.2.2. Teacher Role Transformation:Designer and Guide of Collaborative Inquiry

The integration of artificial intelligence into educational environments not only expands the dimensions of teaching interactions but also drives the transformation of teachers 'roles. In traditional teaching models, educators primarily serve as knowledge transmitters. However, within the framework empowered by intelligent conversational systems, teachers should evolve into designers of learning activities and facilitators of the learning process. This shift moves away from the one-way "teacher asks—student answers" model to establish a two-way interactive structure of "collaborative inquiry—shared reflection." In programming education, teachers should empower students with questioning rights by implementing mechanisms like "question time," "question walls," or "group Q&A sessions." These approaches encourage students to proactively raise questions about programming errors, algorithm selection, code optimization, and other issues. By leveraging AI for real-time feedback and cognitive reinforcement, this approach enhances students' problem-solving awareness and deepens their thinking.

#### 4.2.3. Optimizing Learning Ecology: Dual Improvement of Physical and Cultural Environment

Programming education, as a high-cognitive-load and practice-oriented learning activity, demands robust environmental support. Research indicates that while physical environments don't directly influence instructional interactions, they indirectly shape teaching quality by affecting students' learning motivation and self-efficacy. Educators must therefore design programming-specific learning environments with external support systems. This optimization involves two key dimensions: physical infrastructure and cultural context. First, improving classroom equipment and spatial layouts ensures effective teaching interactions. Second, creating a psychologically safe classroom atmosphere encourages active participation, free expression, and independent exploration. Initiatives like "Bug Sharing Sessions" and "Best Debugging Awards" transform coding errors into learning opportunities, reducing students' fear of failure. Group activities such as "Code Peer Reviews" demonstrate collaborative processes, fostering a sense of belonging and team spirit.

## Acknowledgements

Author Profile: Zhuo Bicheng, male, born in September 1997, of Han ethnicity, native of Ziyang, Sichuan Province. Currently pursuing a master's degree in Educational Technology at the School of Education, West China Normal University, with a research focus on information technology education.

## References

[1] Notice of the State Council on Issuing the Development Plan for the New Generation of Artificial Intelligence [J]. The State Council of the People's Republic of China, 2017(22):7-21.

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0049

[2] Chen Li. The Essence of the Term "Teaching Interaction" and the Analysis of Related Concepts [J]. China Distance Education, 2004(03):12-16+78-79.

[3] Gu Xiaoqing, Wang Chunli, Wang Fei. Has the Role of Information Technology Evolved: A Study on the Impact of Educational Informatization [J]. Research on Electro-education, 2016, (10):5-13.

[4] Zhou Defu. A Brief Analysis of Interaction in Modern Distance Education. Modern Distance Education, 2006, (05):21-23.

[5] MOORE M G. Editorial:three types of interaction [J]. American Journal of Distance Education,1989, 3(2):1-7.

[6] Sun Tianlinzi. Current Status and Optimization Strategies of Online Course Video Interaction Methods [J]. China Distance Education, 2021, (1):57-65.

[7] Xie Changjing. Factors Influencing High-Order Thinking Skills in Junior High School Students' Programming Learning from the Perspective of Classroom Interaction [D]. Southwest University, 2024.