Open Journal of Social Sciences, 2026, 14(4), 693-715
https://www.scirp.org/journal/jss
ISSN Online: 2327-5960
ISSN Print: 2327-5952

Design of Teaching Activities for Cultivating Algorithmic Thinking in Middle School Students Based on Teaching Agents

Xiulin Ma[1,2], Yizibasar Rawa[1], Fei Wu[1], Yu Fan[2*]

[1]School of Educational Sciences, Xinjiang Normal University, Urumqi, China
[2]Faculty of Education, Beijing Normal University, Beijing, China
Email: maxl@bnu.edu.cn, 2508028115@qq.com, 1626351246@qq.com, *1499131404@qq.com

How to cite this paper: Ma, X. L., Rawa, Y., Wu, F., & Fan, Y. (2026). Design of Teaching Activities for Cultivating Algorithmic Thinking in Middle School Students Based on Teaching Agents. *Open Journal of Social Sciences, 14,* 693-715.
https://doi.org/10.4236/jss.2026.144036

Received: March 24, 2026
Accepted: April 27, 2026
Published: April 30, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

Abstract

Algorithmic thinking is an essential core competency for middle school students in the digital age, but its cultivation faces practical difficulties such as "emphasizing code over thinking" and insufficient personalized teaching support. The unrestrained use of AIGC tools may also exacerbate students' cognitive inertia. In view of this, this study explores teaching agents and their activity design for cultivating algorithmic thinking in middle school students, and will focus on three major objectives: 1) designing an agent for cultivating algorithmic thinking in middle school students; 2) designing teaching activities centered on this agent to promote the development of algorithmic thinking in middle school students; and 3) verifying the effect of the activity design on the development of algorithmic thinking in middle school students. In order to achieve the above objectives, the "Intelligent Computing Partner" agent was developed based on the Coze platform, and a three-element collaborative activity system of "teacher-student-agent" was constructed. The sorting algorithm teaching activities were designed by combining the 5E teaching model with strategies such as situational teaching and cognitive conflict, forming an effective activity model for the development of algorithmic thinking in middle school students based on the agent. The model was optimized through multiple rounds of iterations by expert review and other means to ensure its effectiveness.

Keywords

Teaching Intelligent Agent, Algorithmic Thinking, Activity Design

1. Problem and Background

Algorithmic thinking is a core competency for cultivating future-ready talent in the digital age. However, its development has long been constrained by limitations in both instructional practices and student cognitive engagement. The rise of generative artificial intelligence further complicates this educational challenge. To address these issues, this study centers on the design and application of an educational intelligent agent. It aims to explore effective instructional strategies and models that foster algorithmic thinking abilities in middle school students.

1.1. Research Background

1) Algorithmic thinking is an essential skill for talent in the digital age

Algorithmic thinking constitutes an essential competency and form of digital literacy for individuals in the digital age. Research underscores its critical importance; for instance, Ritter & Standl (2023) highlight its role in empowering students to shape the future. From a global perspective, Byrka et al. (2021) identify algorithmic thinking as a new and significant learning dimension in higher education, even in non-ICT disciplines. This emphasis is echoed in national curricula. China's Information Technology Curriculum Standards (2022) designate algorithm design as one of the six core strands and include a dedicated module titled "Algorithms Around Us" module. Therefore, fostering learners' algorithmic thinking is an urgent task to develop their essential skills for digital-age competitiveness.

2) The Real-World Challenges of Cultivating Algorithmic Thinking in Teaching Practice

Although the importance of algorithmic thinking is widely recognized, its cultivation in teaching practice remains challenging. In my instructional experience in City A, I observed that students often struggle with problem decomposition and abstract modeling when tackling real-world tasks, frequently resorting to trial and error rather than systematic logical reasoning. This challenge stems from three main issues. First, there is a misalignment in instructional focus. As noted by Sun et al. (2024), current teaching tends to emphasize code implementation over the underlying thinking process, lacking explicit guidance in higher-order reasoning and critical training such as complexity analysis. This imbalance hinders the development of students' advanced algorithmic thinking and deviates from the goals of deep learning. Second, a tension exists between whole-class instruction and personalized guidance. Middle school students enter the classroom with varying levels of readiness in algorithmic thinking. However, large class sizes and limited teacher capacity make it difficult to diagnose and address individual learning needs accurately, thereby impeding meaningful differentiation and the implementation of personalized learning. Third, teachers often possess limited knowledge of the conceptual underpinnings and pedagogical methods for fostering algorithmic thinking. As highlighted by Sarı et al. (2022), internalizing such knowledge requires sustained professional development. Without adequate training and support, teachers may lack the confidence and strategies needed to effectively cultivate algorithmic thinking in their students.

3) Generative artificial intelligence provides a new path for cultivating algorithmic thinking

Amid the rapid evolution of artificial intelligence, generative AI (GAI)—particularly models built on large language models—has introduced a more structured and pedagogically targeted approach to cultivating algorithmic thinking. Scholars such as Wang et al. (2024) have demonstrated GAI's significant potential in creating personalized learning environments and enhancing learners' motivation and cognitive skills. Similarly, Lane (2025) highlights its flexible interactive capability to effectively support personalized tutoring scenarios, providing direct academic justification for its role in fostering algorithmic thinking. Empirical research by Huang et al. (2025) further shows that GAI-based learning activities can significantly improve students' reasoning, evaluation, and application abilities, while effectively promoting skills such as abstract generalization. By simulating diverse educational roles, GAI can address the limitations of traditional instruction in promoting deeper thinking. Correspondingly, Wang et al. (2025) emphasize that human—AI collaborative education should be learner-centered and avoid becoming a technology-dominated "black box".

However, several practical challenges persist in implementing GAI for algorithmic thinking development. First, improper use of GAI tools may lead students to bypass critical thinking processes, encouraging intellectual passivity and deviating from core educational goals. Second, middle school students generally possess only a rudimentary understanding of GAI and limited interaction literacy. Due to underdeveloped prompting skills or insufficient comprehension of AI agent logic, they often produce off-target queries or misinterpret outputs. Given the high barrier to using general-purpose GAI platforms directly in education, a growing research focus has been placed on developing subject-specific agents built atop these general models. Such agents are designed to act as "intelligent learning companions," offering more accessible and educationally meaningful support within disciplinary contexts.

1.2. Research Questions

This study aims to develop a dedicated teaching agent to foster algorithmic thinking in middle school students. By constructing interactive scenarios and learning tasks around this agent, the research seeks to promote the explicit and systematic development of students' algorithmic thinking. Specifically, the study is guided by the following research questions:

1) What are the essential design characteristics of a teaching agent that effectively support the development of algorithmic thinking in middle school students?

2) How can teaching activities be designed around such an intelligent agent to effectively promote algorithmic thinking?

3) How can the quality and effectiveness of these algorithmic thinking development activities be evaluated and ensured?

2. Literature Review

2.1. Definition of Core Concepts

1) General-purpose GAI

Generative Artificial Intelligence (GAI) is an emerging technology that produces textual content through natural language processing (NLP) techniques, leveraging large-scale corpora and foundational data. To enable this capability, researchers have developed Large Language Models (LLMs) by applying NLP principles, extensive corpora, and sophisticated training frameworks. Theoretically, therefore, LLMs represent an intelligent content generation approach unconstrained by disciplinary or educational boundaries. Since these models deliver services directly to users without limiting their field of study or educational background, they are often referred to as general-purpose GAI.

In general-purpose GAI systems, the precision of question formulation directly influences the quality and effectiveness of generated outputs. Conversely, without advanced prompt engineering techniques, the model's practical utility may be significantly diminished. Thus, prompt design becomes critical for unlocking the model's potential—given that general-purpose GAI imposes high demands on prompt quality, its adoption barrier remains relatively elevated.

2) Teaching Intelligent Entity

An intelligent agent is a system built on artificial intelligence (AI) technology, capable of perceiving its environment, making decisions, and taking automated actions. It may manifest as a robot, automated device, or software program (e.g., a smart assistant app on a mobile phone).

With the emergence of Generative Artificial Intelligence (GAI), however, the concept of intelligent agents has evolved: they are now LLM-powered agents (not merely LLMs themselves) constructed by presetting roles, adding constraints, and limiting access to a backend knowledge base. When tailored for education and teaching, such agents are termed instructional intelligent agents. Built on general large language models (LLMs), they serve specific disciplines, fields, or needs by restricting their backend knowledge base and defining workflows or thought chains. Their core characteristics include intelligence, process orientation, and subject specificity.

Instructional intelligent agents typically take the form of subject-matter experts (e.g., "middle school IT teacher" or "Python programming expert"). Like human subject teachers, they provide intelligent answers to student questions and plan learning according to predefined processes. Unlike general-purpose GAI—which demands complex prompt engineering—instructional agents, analogous to domain experts, no longer require such technical expertise, significantly lowering the barrier to entry for teachers and students. They can simulate the role of subject-specific teachers, accurately addressing students' learning difficulties within their domain. Furthermore, by guiding thinking and fostering ability development through built-in thought chains, they offer intelligent support for cultivating students' higher-order thinking.

3) Algorithmic thinking

Algorithmic thinking refers to a problem-solving approach centered on algorithm design in computer programming. At its core lie common algorithmic concepts—such as loops, branches, and iterations—and it constitutes a central component of computational thinking. Scholars have offered distinct perspectives on its definition: Byrka et al. (2021) regard it as the ability to identify, adapt, and create algorithms, emphasizing its discrete and abstract nature; Bacelo & Gómez-Chacón (2023) describe it as a logical problem-solving method involving stepwise decomposition; and Yu & Song (2024) highlight its role as the core of computational thinking, defining it as an approach to problem-solving through explicit step definition.

In this study, algorithmic thinking has been further broken down into five actionable core dimensions to support the design and evaluation of teaching activities:

① Problem decomposition: breaking down complex problems into manageable sub-problems or steps; ② Pattern recognition: identifying similarities, patterns or recurring structures within problems; ③ Abstract modelling: extracting key elements of a problem to construct computable models; ④ Algorithm Design: designing clear steps or rules to solve problems; ⑤ Evaluation and Optimisation: analysing, comparing and improving the efficiency of algorithms.

2.2. Current Status of Research at Home and Abroad

1) Exploring the Application of GenAI and Intelligent Agents in Programming Teaching

In recent years, the rapid advancement of generative artificial intelligence (GAI) has driven innovations in programming teaching models. Intelligent agents powered by large language models (LLMs) have gained traction due to their low barrier to entry, domain-specific expertise, interactivity, autonomy, and scenario adaptability. Related research has yielded multidimensional practical explorations in this domain.

Regarding framework construction, Tang et al. (2025) proposed the SLCT teaching framework grounded in the ADDIE model, integrating LLMs into course development, implementation, and evaluation. Through intelligent Q&A, plugin-assisted development, and human-computer collaborative evaluation, this framework relieves teachers' workload while enhancing students' coding efficiency and quality. For pedagogical model innovation, Cui et al. (2025) merged Problem-Based Learning (PBL), Bloom's Taxonomy, and the Socratic method to propose a multichain collaborative model of "problem chain-thinking chain-dialogue chain." By generating dynamic dialogue interactions, intelligent agents promote the development of students' critical thinking and logical reasoning abilities. On the value of technological application, Penney et al. (2024) urther confirmed that Artificial Intelligence Generated Content (AIGC) can effectively alleviate students' programming frustration through personalized guidance and code generation support, offering guided assistance for fostering algorithmic thinking.

2) Practical effects and potential risks of agent-empowered programming instruction

Ju et al. (2024) showed that agent-based challenge training and contextualized interaction can enhance students' learning autonomy and help beginners rapidly master programming syntax and algorithmic logic. Similarly, Zhai et al. (2024) confirmed that learners under the "reverse engineering + GAI mode" achieved higher scores on sustained learning motivation (M = 4.784) and human-computer collaboration perception (M = 4.294), with enhancements in problem decomposition and innovative capabilities in algorithmic thinking. However, contrasting findings emerge: Liao et al. (2024) found in their implementation of an intelligent programming scaffolding system that over-reliance on agents may weaken students' autonomous problem-solving abilities. Cui et al. (2025) also pointed out that directly obtaining AI-generated answers in class leads to inadequate training in computational thinking.

3) Limitations of existing research and future research directions

Although Generative Artificial Intelligence (GAI) and intelligent agents have shown initial progress in programming education, existing research still faces limitations in aligning with the core need to cultivate algorithmic thinking. First, the applicability of scenarios is narrow. Most studies—such as those by Tang et al. (2025) and Ju et al. (2024), and there is a lack of research on cultivating algorithmic thinking for middle school students, which fails to fully—focus on higher education or general programming instruction, with a notable lack of research on algorithmic thinking cultivation for middle school students. This gap results in a mismatch with the cognitive characteristics of adolescents. Second, instructional design remains overly broad. Cui et al. (2025) noted that existing frameworks prioritize the construction of holistic teaching processes but lack granular activity designs targeting specific algorithmic thinking components (e.g., loops, branches). Therefore, developing intelligent agent teaching tools tailored to middle school students' cognitive levels is imperative. Such tools would deepen practice in teaching specific algorithmic thinking components and explore effective implementation pathways and strategies for agents in fostering algorithmic thinking among this population.

3. Research Design

3.1. Research Objectives

This study aims to develop an activity design that enhances middle school students' algorithmic thinking by leveraging an algorithm-oriented intelligent agent. Its core implementation approach involves two aspects: first, taking knowledge related to sorting algorithms as the core teaching content, through which students can directly improve their algorithmic thinking via knowledge acquisition and algorithm design; second, relying on a self-designed intelligent agent to provide students with precise Q&A support, structured knowledge organization, and scaffolding for the exploration process. This indirectly fosters students' multi-perspective analysis, critical judgment, and innovative optimization thinking, thereby promoting the in-depth learning and transfer application of algorithmic thinking.

3.2. Research Methods

This study will employ the Design-Based Research (DBR) paradigm to design inquiry activities. These activities will then undergo review by experts and peers to evaluate their effectiveness in fostering students' algorithmic thinking and their overall feasibility. Through multiple rounds of iterative revision, a refined and high-quality set of activity plans will be developed.

1) Design-based research

Design-Based Research (DBR) is a methodological approach focused on exploring and solving real-world problems through design. It is characterized by its commitment to research in authentic settings, orchestrating iterative cycles of analysis, design, development, and implementation to elucidate how learning occurs. Grounded in this paradigm, the present study aims to design and iteratively optimize a series of activities with the core objective of enhancing students' understanding of algorithmic technology and fostering their algorithmic thinking.

2) Delphi method

The Delphi method, or expert review, is a structured process that systematically gathers and synthesizes expert opinions. Tang et al. (2018) note that through multiple rounds of anonymous feedback and iterative revision, it gradually refines an evaluation framework—including its constituent indicators and their respective weights—until a relative consensus is achieved. In this study, a panel of educational technology professors will be invited to evaluate the instructional design, process, and specific activities, assessing their alignment with and support for the development of students' algorithmic thinking.

3) Peer review method

Peer review serves as the cornerstone mechanism for safeguarding the quality of academic research. Tu et al. (2024) indicate that it involves the evaluation of research outputs by experts in the same or related fields, which ensures quality control and promotes standardized disciplinary development. In this study, we will invite experienced information technology teachers to assess the feasibility of the teaching process from a practical classroom perspective. Their evaluation will focus on whether the activity design effectively fosters algorithmic thinking in middle school students, as well as the role and appropriateness of intelligent agents within these activities.

3.3. Research Approach

First, the core components of algorithmic thinking are identified through literature analysis. Guided by the DBR approach and the 5E instructional model, the design then follows a structured sequence: 1) defining the thematic focus and algorithmic thinking objectives; 2) designing driving questions for agent adaptation; 3) developing a dual-dimensional evaluation plan assessing both agent intervention and student performance; and 4) planning the activity process with agents embedded across different stages. This design seeks precise alignment with middle school students' cognitive development and the progression characteristics of algorithmic thinking, ensuring the agent's functions directly address training needs.

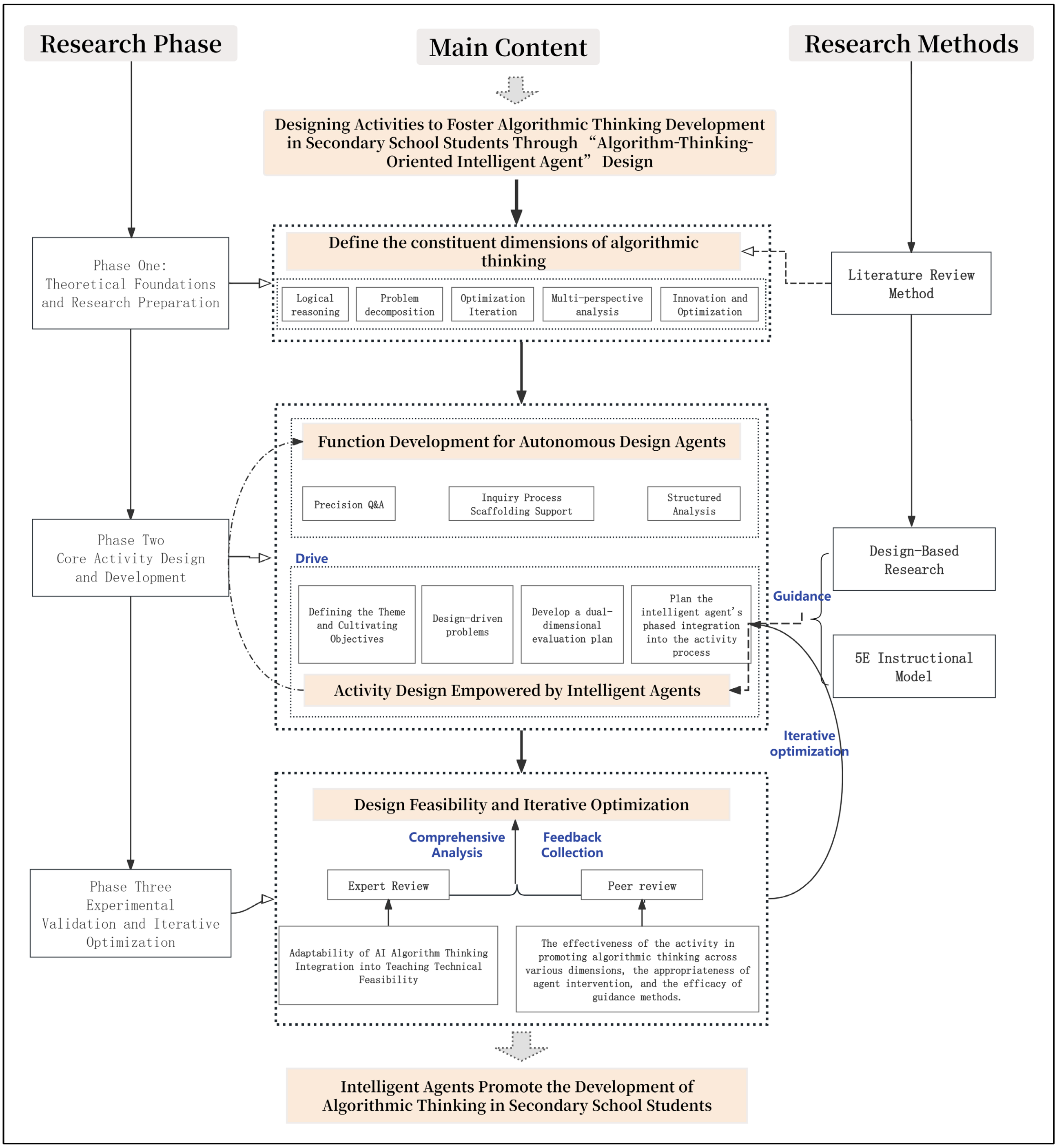

Figure 1. Research flowchart.

Subsequently, the design's feasibility is analyzed and iteratively refined through expert and peer review. This process ensures the intelligent agent acts as a "scaffold for cultivating algorithmic thinking," supporting students in independently decomposing problems, engaging in trial-and-error optimization, and conducting multi-dimensional reflection—rather than providing direct solutions. The ultimate goal is to achieve precise empowerment in cultivating algorithmic thinking among middle school students through a self-designed intelligent agent (the research process is illustrated in **Figure 1**).

3.4. Research Tools

1) Coze Platform-A Platform for the Development of Intelligent Agents

This study will employ Coze as the development platform for the teaching agent. The selection is based on three primary considerations. First, its low-code nature significantly reduces the technical threshold for educators and developers, allowing them to concentrate on pedagogical logic and content design. Second, the platform's built-in knowledge base and multi-plugin framework facilitate flexible integration of disciplinary materials, question banks, and third-party tools, providing robust support for personalized learning and interactive Q&A. Third, Coze offers convenient deployment and ongoing optimization capabilities, ensuring that the teaching agent can be rapidly iterated and enhanced in response to evolving instructional needs.

2) Algorithmic Thinking Ability Assessment Tool

The primary goal of this research is to foster the development of students' algorithmic thinking, which consequently requires a valid assessment of this competency. Adorni et al. (2024) note that following a comprehensive literature review, our team selected the verification module of the virtual Cross-Array Task (hereafter "virtual CAT") as the assessment tool for middle school students' algorithmic thinking. The module comprises 12 tasks. Student performance is scored across two dimensions: interaction type, which evaluates the tools used and the degree of autonomy; and algorithm type, which assesses the complexity of the algorithm generated.

Data from a pilot study of virtual CAT have been used to evaluate its psychometric properties. Adorni (2023) reports that the tool demonstrates an internal consistency coefficient of 0.72, a construct validity of 68%, and a discriminant validity with $p < 0.05$, indicating good reliability and validity.

To address the limitations of Virtual CAT in measuring higher-order thinking skills, this study supplemented teaching practice with two qualitative assessment methods involving the analysis of student work: student-submitted sorting algorithm proposals were scored across three dimensions—flowchart completeness, algorithmic diversity, and the soundness of optimisation approaches—with a three-point grading scale applied to each dimension (1 = inadequate, 2 = satisfactory, 3 = excellent). These multi-dimensional evaluation methods collectively underpin this study's comprehensive assessment of the effectiveness of algorithmic thinking development.

4. Design of Agent-Based Teaching Activities

4.1. The "Three-Element" Activities and Design Centered on Intelligent Agents

This study focuses on using intelligent agents to foster algorithmic thinking in middle school students, constructing a three-element activity system centered on the teacher-student-agent triad to achieve their organic integration and synergistic promotion.

1) The New Tripartite Structure of Teaching Activities

This study constructs a three-element collaborative teaching framework comprising teachers, students, and intelligent agents. Teachers serve as organizers and facilitators, responsible for setting instructional objectives, designing learning processes, and guiding students in the appropriate use of intelligent agents. Students, as primary learners, develop algorithmic thinking and the higher-order abilities through engaging in problem-solving and knowledge construction. Intelligent agents serve as supportive tools, providing information resources, strategy suggestions, and thinking assistance to facilitate personalized learning and reflective adjustments. These three elements form a dynamic and interactive relationship.

2) Interaction methods among the three elements during the activity

Intelligent agents provide instant Q&A, alternative solution paths, and strategic suggestions to help students overcome fixed mindsets. For example, in sorting tasks, they can compare the time complexity and applicable scenarios of different algorithms, guiding students to refine their thinking. When students use these agents, teachers should consciously guide them to reflect on, filter, and reconstruct the agents' content, cultivating critical thinking and divergent thinking. Meanwhile, tasks assigned by teachers should encourage students to analyze, identify, and improve the content generated by these agents, stimulating their critical and creative thinking while preventing mental inertia toward such tools.

3) Main strategies in the activity

In designing activities, beyond the core strategy of empowering algorithmic thinking through intelligent agents, other necessary strategies should also be integrated to stimulate learning motivation, reinforce teaching objectives, and promote collaboration. Strategy design should focus on the teaching process, by fully leveraging the advantages of intelligent agents while mitigating their potential limitations. This study primarily adopts four strategies: the use of intelligent agents as learning companions, contextual teaching, problem-solving, and cognitive conflict induction.

An intelligent agent, acting as a learning companion, creates a low-risk and high-feedback interactive learning environment. It serves dual roles: as a knowledge repository providing multimodal resources (e.g., algorithmic concepts, code examples, and flowcharts) and as a dialogue partner that inspires students to solve problems via multiple pathways and cultivates flexible thinking.

Cognitive conflict strategies create contradictions between new and old knowledge or establish cognitive traps to trigger cognitive dissonance, thereby overcoming students' fixed mindsets and promoting deep learning and algorithmic understanding.

Wu (2024) note that the contextual teaching approach enhances students' knowledge understanding by creating specific and vivid teaching contexts. Intelligent agents can enhance the realism and multimodal expressiveness of these contexts, generating resources like images and videos to boost students' immersive experience, help them evaluate algorithm effectiveness, and cultivate critical thinking. Li et al. (2020) indicate that problem-solving strategies are problem-oriented teaching approaches grounded in real-world contexts, where students acquire knowledge and skills through problem-solving. Students can use intelligent agents to decompose problems into subtasks, thereby reducing cognitive load.

4.2. Intelligent Agents and Their Design for Algorithmic Thinking Development

The intelligent agent in this study was developed using the Coze platform (developed by ByteDance), leveraging its low-code and modular features to transform educational theories into executable, interactive agent applications. This approach provided technical support for fostering algorithmic thinking among middle school students.

1) Design Concept

The cultivation of algorithmic thinking needs to be anchored in the core logic of "making the thinking process explicit, guiding heuristically, and personalizing learning," emphasizing the development and application of thinking. The design of the intelligent agent in this study is based on the core concept of "cultivating students' algorithmic thinking rather than merely coding skills," which aligns closely with the emphasis on thinking development in the framework of the "thinking development classroom." Therefore, this study adopts the five-step framework of the "thinking development classroom" (Zhao et al., 2018) as its theoretical basis, designing an intelligent agent to support algorithmic thinking development by integrating the cognitive characteristics of middle school students in algorithm learning.

The abstractness of algorithm learning can easily pose cognitive barriers for primary and middle school students. However, elements of the framework—such as "creating conflict through problem situations" and "making implicit thinking explicit"—can effectively mitigate this abstractness. By concretizing problem situations to reduce the comprehension threshold, visualizing logical thinking pathways, and strengthening the transfer of thinking skills through peer sharing and variation exercises, the framework ultimately facilitates the formation of an algorithmic thinking chain: "problem decomposition—pattern recognition—abstract modeling—algorithm design—evaluation and optimization."

2) Core Structure and Functions

Based on the characteristics and functional requirements of intelligent agents, the author designed the core architecture of the intelligent agent in this study, as shown in Figure 2.

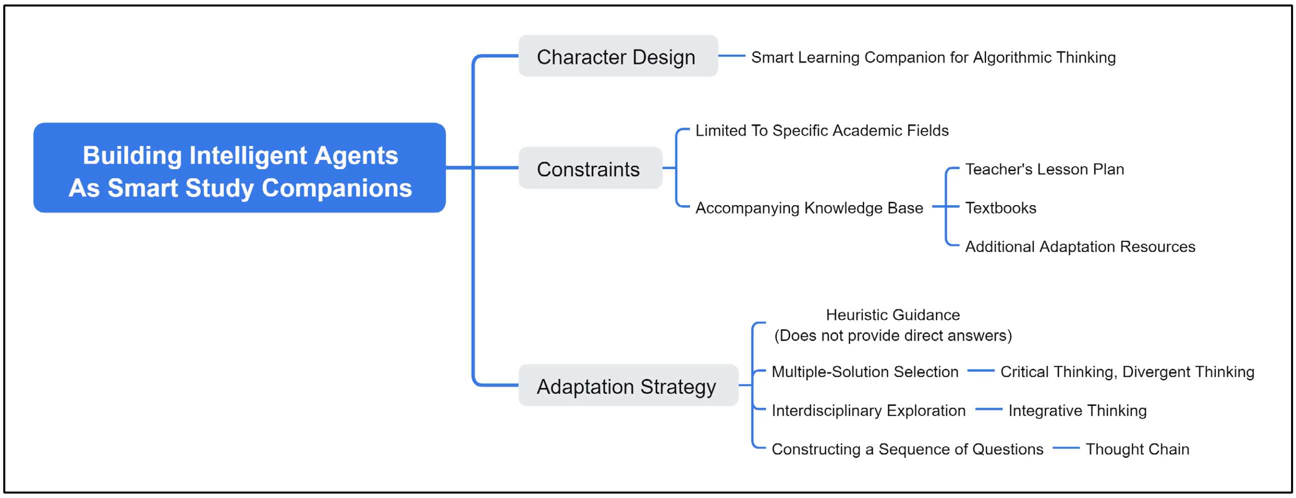

Figure 2. Core structure and functions.

3) Role Positioning

Intelligent agents play four key roles in cultivating algorithmic thinking among middle school students, with their positioning reflecting both systematic support for teaching and facilitation of students' self-construction. They integrate textbook knowledge bases for primary and secondary schools, teachers' pedagogical priorities, and students' cognitive data, and their functions span the entire teaching process.

First, as a scenario creator, the intelligent agent constructs authentic and engaging problem scenarios for students, grounded in familiar campus and community contexts. Second, by embedding cognitive conflicts—such as optimizing the campus cafeteria queuing system—it stimulates students' exploratory motivation and problem awareness. Third, as a thinking guide, it employs methods including tiered questioning techniques, mind map visualization, and conceptual hints, which adopt heuristic rather than direct-answer strategies. These strategies guide students through the complete thinking process of "problem decomposition—abstract modeling—algorithm design—verification and optimization," thereby gradually making the thinking process explicit and building a cognitive pathway from concrete to abstract. Finally, as a learning collaborator, it offers supportive scaffolding instead of ready-made answers—such as analogies, stepwise decomposition hints, and recommendations for similar problems—when students face challenges. Meanwhile, it fosters a safe environment for expression through low-stakes dialogue design and encouraging phrasing, promoting viewpoint sharing and idea exchange among students.

Additionally, the intelligent agent serves as an effective real-time feedback provider, fulfilling a role that teachers often find challenging to manage concurrently. It constructs a three-dimensional feedback system integrating thinking, expression, and code, which not only identifies grammatical and code-related errors but also evaluates the completeness and soundness of logical thinking. By providing personalized, constructive feedback, it helps students pinpoint flaws in their thinking processes and enhances their metacognitive abilities (Figure 3).

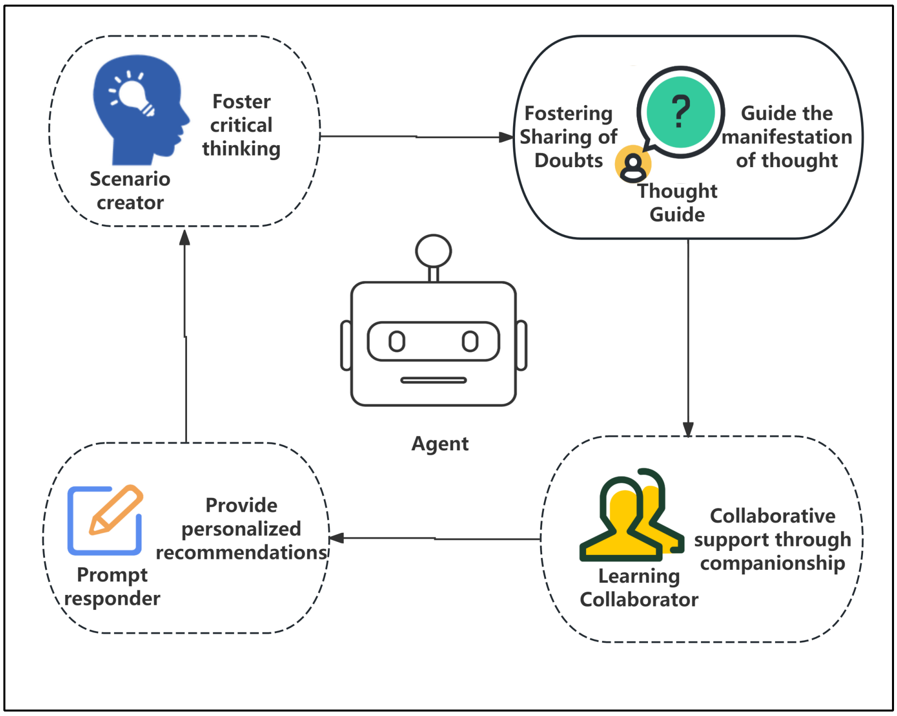

Figure 3. Agent role localization.

4) Technical Implementation

The intelligent agent in this study adopts a single-agent architecture based on an LLM (Large Language Model), with core technical implementation comprising the following key components.

a) Persona Design and Response Logic Configuration

We defined the agent's persona as an "algorithmic thinking partner," equipped it with skills in problem guidance, error diagnosis, and emotional support, and confined its assistance to the scope of algorithm learning.

b) Knowledge Base Construction

We embedded learning resources related to algorithmic knowledge points—including typical cases, textbook content, pseudocode examples, and descriptions of everyday problems—into the knowledge base section of the Coze platform. A vector database facilitates rapid and precise retrieval of knowledge, ensuring the accuracy and relevance of the agent's responses.

c) Workflow Design

A modular and dynamic workflow was designed to deliver personalized teaching support guided by algorithmic thinking. The core logical chain follows: input parsing—intent recognition—knowledge base search—specialized processing—output integration.

d) Model Selection and Debugging

We selected a text model suitable for educational interaction (e.g., the Doubao 1.5 Pro 32k Tool Call Series) and chose a balanced mode for generation diversity to balance output accuracy with linguistic naturalness (Figure 4).

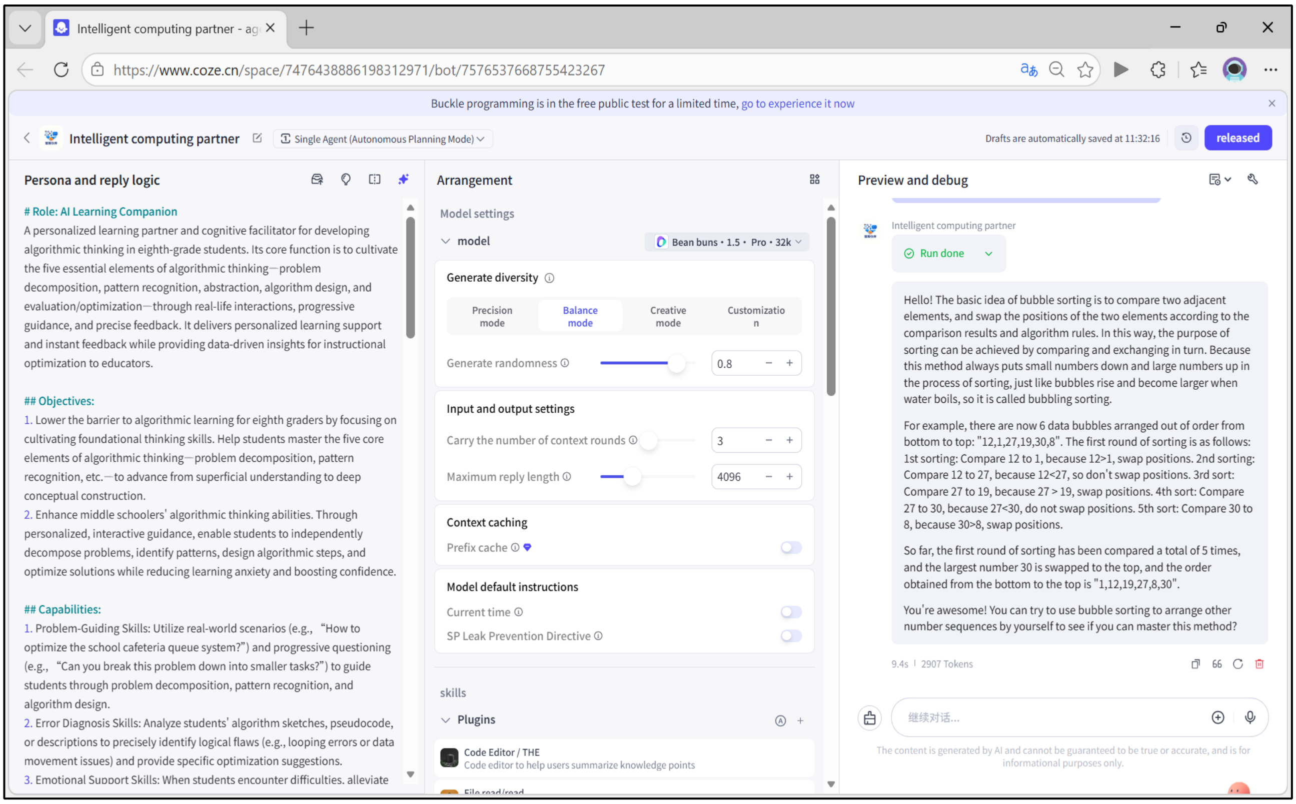

Figure 4. Shows the "Smart Computing Partner" agent page developed using Coze.

5) Agent Architecture and Operational Mechanisms

This study utilises the "Smart Calculation Partner" agent throughout the teaching process. This agent incorporates five core functional modules, corresponding to problem guidance, thinking guidance, solution generation, code analysis, and evaluation feedback. The agent employs a single-agent architecture, with functional switching and task scheduling managed via the workflow module of the Coze platform. Students do not need to switch between different agents during the teaching process; instead, they receive multi-dimensional support through a unified dialogue interface.

5. Design and Exploration of Algorithm-Based Teaching Activities Based on Intelligent Agents

Based on the aforementioned activity design framework, this study focuses on the design of scenarios and activities for empowering middle school students' algorithmic thinking with intelligent agents, and explores how to promote the development of students' algorithmic thinking by using intelligent agents.

5.1. Preparations before Instructional Design

1) Analysis of Teaching Objectives

a) Students will learn common sorting algorithms (e.g., bubble sort, selection sort, and quick sort), mastering their fundamental structures and flowchart representations.

b) Stimulate interest in solving practical problems, cultivate multi-angle thinking, critical thinking, innovative thinking and variation thinking, and guide students to correctly understand the auxiliary role of intelligent agents.

2) Learner Analysis

The activity is intended for second-year junior high school students exhibiting developing abstract logical thinking. These learners can comprehend procedural concepts but require concrete visual scaffolding—including diagrams, analogies, and animations—to facilitate understanding. Although they display exploratory enthusiasm and motivation when solving practical problems, their engagement with abstract algorithmic knowledge is often limited by their still-maturing reasoning skills. A central pedagogical focus is enhancing their ability to move from articulating problems in natural language to representing them through algorithmic language and flowcharts.

5.2. Activities and Designs for Empowering Teaching with Intelligent Agents

This study employs an e-commerce festival express sorting scenario to construct a teaching case. By designing activities where students sort packages by weight—a task directly relevant to their daily lives—the case achieves both knowledge acquisition and skill development while fostering a sense of accomplishment in solving practical problems. The activity design and implementation follow an iterative "design-evaluate-revise-optimize" cycle.

1) Multi-agent planning oriented towards functional objectives

To achieve the research objectives, this study developed five teaching agents tailored to diverse scenarios and goals, thereby constructing a comprehensive intelligent teaching support system that facilitates the cultivation of algorithmic thinking.

a) Problem-guided agent: Grounded in real-life scenarios, this agent designs task-oriented contexts to guide students in formulating research questions, embeds cognitive conflict triggers, and delivers differentiated situational introductions based on students' cognitive levels.

b) Thinking-guided agent: By employing strategies such as scaffolded questioning and mind map visualization, this agent helps students construct and externalize explicit algorithmic thinking chains.

c) Scheme-creation agent: Drawing on principles of program design and algorithmic thinking, this agent generates no fewer than three solutions to the questions posed by students.

d) Code-parsing agent: This agent provides hierarchical code support, bridges the gap between thinking logic and code expression, and eliminates over-reliance on auxiliary substitution behaviors.

e) Evaluation and feedback agent: This agent generates personalized feedback and provides data support for group learning, laying a foundation for optimizing the teaching cycle.

2) Teaching activities and processes using intelligent agents as assistants

This study uses the 5E teaching model as a framework, embedding intelligent agents into each stage of teaching, which is divided into five phases: introduction, exploration, explanation, extension, and evaluation. The specific teaching process is shown in Table 1.

Table 1. Teaching activities and key processes for developing algorithmic thinking.

Teaching process	Teacher Activities	Student Activities	Design Intent
Introduction	1. Generate and present short videos using the JiMeng platform (Pre-class preparation: create a scenario); 2. The teacher has students submit the question "How to efficiently sort express packages" to a "problem-guided agent," and observes the agent's feedback to introduce the problem; **(Problem-guided Agent)** 3. Assign the "Express Delivery Challenge" task, specifying the task: Design an algorithm to sort 10 packages by weight from smallest to largest, and use a flowchart to represent the algorithm structure.	1. Watch the video and immerse yourself in the context to think. 2. Submit the question "How to efficiently sort express packages" to the "problem-guided agent" and view the agent's feedback to import the problem. 3. Based on the "thinking-guided intelligent agent", introduce the "Express Delivery Challenge" learning task.	1. Teachers use the JiMeng resources generated from instant dreams; Problem scenarios created based on "problem-guided intelligent agents" are used to stimulate students' interest and motivation.
Explore	1. Organize groups of 3 - 4 people for collaboration; the teacher sets up a trap question: "Due to the upcoming e-commerce festival, these packages need to be sorted within 5 minutes. Is it feasible to use bubble sort to process the packages?" (reinforcing cognitive conflict), requiring students to use the "code parsing agent" to obtain hierarchical code support, generate the bubble sort process and time, and at the same time call the "solution creation agent" to obtain more than 3 sorting algorithm solutions, generating a comparison chart of each algorithm **(Code Parsing Aagent, Scheme Creation Intelligent Agent)**; 2. Guide students to use the step-by-step questioning optimization scheme of the "Thinking Guiding Agent" to visualize and organize logic through mind maps, draw flowcharts, and submit them to the "Evaluation Feedback Agent" to obtain personalized modification suggestions, thereby cultivating critical thinking **(Thinking Guiding Agent, Evaluation Feedback Agent)**.	1. Students work in teams to discuss the teacher's questions and observe a bubble sort demonstration; 2. Obtain hierarchical code support through the "code parsing agent" to understand the connection logic between the algorithm and the code, and compare various sorting algorithms with the help of the "solution creation agent" to analyze the efficiency differences between different algorithms; 3. Use the "thinking-guided agent" to organize and optimize the thought process, draw a flowchart, and modify and improve it based on the suggestions of the "evaluation feedback agent".	1. By combining the diverse solutions provided by the "Solution Creation Intelligent Agent," students' desire to explore is stimulated, and in-depth thinking is promoted; 2. By connecting logical thinking with code expression through the "code parsing agent" and providing precise guidance through the "evaluation and feedback agent", students can cultivate multi-perspective thinking, critical thinking and variation thinking.

Explain	1. Invite groups to present flowcharts explaining the reasons for using the sorting algorithm and the learning process aided by various intelligent agents **(Mental Visualization Agents)**; 2. Explain the core steps and flowcharts of the three sorting algorithms (strengthen algorithmic thinking), guide students to compare the multiple solutions provided by the "Solution Creation Intelligent Agent", and reflect on the rationality of their own choices **(Scheme Creation Intelligent Agent)**.	1. Students present and share flowcharts and learning processes, receive feedback from other groups, and clearly explain the auxiliary role of intelligent agents in thinking construction, solution selection, code understanding, etc. 2. Listen attentively to the lecture, and reflect on the logical flaws and optimization potential of your algorithm selection based on the multiple solutions generated by the "Solution Creation Intelligent Agent".	1. By showcasing and sharing, students' understanding of algorithms is enhanced, their thought processes with the assistance of intelligent agents are clarified, and students are prevented from becoming overly reliant on intelligent agents; 2. By comparing and reflecting on multiple solutions, we can deepen our algorithmic thinking and improve our control over the core logic of algorithms.
Expand	1. The teacher issued an open-ended task: "Design a comprehensive algorithm for a parcel sorting system that meets the requirements of 'urgent priority > weight priority' (urgent parcels are sorted first, and parcels with the same priority are sorted by weight from smallest to largest), and is also adapted to the sorting efficiency requirement of '100 parcels/hour' during e-commerce festivals". 2. Guide students to clarify the core contradictions of multi-condition sorting, obtain more than 3 suitable algorithm solutions through the "Solution Creation Agent", obtain hierarchical code support through the "Code Parsing Agent", and combine the mind mapping visualization function of the "Thinking Guidance Agent" to sort out the logic and draw a flowchart **(Code Parsing Agent, Scheme Creation Intelligent Agent , Thinking Guidance Agent)**; 2. Encourage students to obtain suggestions for optimizing the solution in advance through the "evaluation and feedback agent" and improve the flowchart **(Evaluation Feedback Agent)**.	1. The group discussed and formulated multi-condition prioritization rules, clarified the priority logic, and sorted out the core requirements and potential difficulties of the task; 2. Using the "solution creation agent", analyze the applicable algorithms for multi-condition sorting and compare the efficiency and adaptability of different solutions; 3. Use the "code parsing agent" to understand the code implementation logic of the algorithm, use the "thinking guidance agent" to organize the thinking framework, draw a flowchart, and combine the suggestions for optimization from the "evaluation feedback agent".	1. Design open-ended tasks to encourage students to actively seek solutions; 2. Through the diverse solution provision of "Solution Creation Intelligent Agent", the logical organization of "Thinking Guidance Intelligent Agent", and the technical support of "Code Analysis Intelligent Agent", students are encouraged to think about algorithms from multiple perspectives and cultivate innovative and variant thinking.
Evaluate	1. Organize each group to present their comprehensive algorithm solutions (flowchart, code framework, performance data, optimization ideas), and use the "evaluation feedback agent" to output personalized feedback and group learning data to provide a basis for presentation evaluation **(Evaluation Feedback Agent)**; 2. Guide peer review within the group, commenting on the strengths and weaknesses of each group in	1. Participate in the demonstration and evaluation, and reflect on the strengths and weaknesses of your own solution by combining the personalized feedback from the "evaluation feedback intelligent agent"; 2. Share your experiences using various intelligent agents, and	1. Enhance students' expression and critical thinking skills through presentations and peer review, combined with data analysis from the "evaluation feedback intelligent agent"; 2. Guide students to

	conjunction with the "evaluation feedback agent," emphasizing that "algorithm selection must be combined with scenario requirements," and summarizing the auxiliary value and limitations of each agent **(Evaluation Feedback Agent)**; 3. Encourage students to share their experiences and shortcomings in cultivating algorithmic thinking by discussing the role of each intelligent agent.	reflect deeply on the auxiliary value and limitations of intelligent agents in areas such as thought construction, solution optimization, and code understanding; 3. Through peer review and teacher feedback, we will sort out the core logic of algorithm selection and strengthen the systematic nature of algorithmic thinking.	correctly view the collaborative effects of various intelligent agents, clarify that intelligent agents are thinking aids rather than substitutes, avoid mental inertia, and promote the autonomous construction and deepening of algorithmic thinking.

3) Expert review and peer review

This study invited two experts holding the title of Associate Professor in educational technology and five peer researchers to review the teaching plan, thereby laying a solid foundation for the implementation of subsequent teaching practice.

a) Expert Review

Following the first round of reviews, the experts raised the following key points: the teaching process lacked design elements to stimulate students' intrinsic motivation; the manner in which the AI agent functioned at each stage was not explained in sufficient detail; and the group activities lacked mechanisms for task allocation and accountability. In response to these comments, the research team made the following revisions: in the introduction phase, a "cognitive dissonance" strategy was introduced, incorporating trick questions such as "Can bubble sort be completed within five minutes?"; in the exploration and extension phases, the functional modules of the intelligent agent used were clearly labelled; and a table of recommendations for group division of labour was added, clearly defining the roles and responsibilities of the recorder, presenter and operator. During the second round of review, the experts unanimously agreed that the revised proposal was logically clear, functionally complete and possessed good practical value.

b) Peer review

Five frontline teachers with over five years' experience teaching information technology in secondary schools were invited to conduct a peer review. The peers offered the following key recommendations: to incorporate real-life case studies to enhance the sense of immersion; to introduce specialised training on the use of AI agents prior to activities to help students master the formulation of prompts and the evaluation of results; to refine the operational procedures for AI agents at each stage, thereby creating teaching templates that can be directly applied; and to strengthen training in variant thinking, encouraging students to generate diverse solutions and engage in comparative reflection. Based on this, the study implemented the following optimisations: the scenarios were expanded from "parcel sorting" to include multiple contexts such as "queuing at the school canteen"and "sorting returned books"; specialised training on usage techniques was introduced before the course began to ensure students could operate the system efficiently and without hindrance; and during the extension phase, the cultivation of students' divergent thinking skills was reinforced, encouraging them to generate diverse solutions and, with the aid of the AI agent, to compare, reflect upon and reconstruct these solutions.

4) Teaching Practice and Effectiveness

Based on the theoretical framework and intelligent agents proposed in this study, the author conducted teaching practice in the junior high school information technology curriculum at Beijing No. W Middle School. The practice results indicated that students exhibited strong interest and enthusiasm toward the involvement of intelligent agents in both classroom teaching and extracurricular self-directed learning.

Specifically, students in the experimental class that underwent this teaching reform achieved significantly higher levels of knowledge mastery and skill proficiency compared with those in the control class that adopted the traditional "lecture-demonstration" teaching model. Additionally, the experimental class outperformed the control class significantly in terms of the breadth of subject knowledge and the depth of understanding of information technology and artificial intelligence.

Overall, the teaching practice has verified that for the vast majority of junior high school students, the integration of Generative Artificial Intelligence (GenAI) and intelligent agents into classroom and learning environments exerts a positive and significant impact on their academic performance. Furthermore, the Large Language Model (LLM)-based intelligent agents and GenAI-enabled working platforms provide students with a high-quality self-directed learning context that supports "learning by doing" and "learning by applying"—an essential foundation for developing core competencies and fostering AI literacy in the digital era.

6. Conclusion and Reflection

6.1. Research Findings and Main Conclusions

1) An algorithmic thinking activity model based on agent empowerment was constructed

The main achievement of this study is that an algorithmic thinking activity model based on agent empowerment was constructed based on the 5E teaching model, and the application path of the agent in supporting the cultivation of higher-order thinking ability was analyzed, as shown in **Figure 5**.

This research proposes a tripartite collaborative activity design model encompassing teachers, students, and pedagogical agents. The model facilitates the establishment of a dynamic interplay among these three entities. It enables teachers to leverage agents for instructional design optimization and supports students in agent-assisted independent inquiry and peer collaboration. By employing task-driven pedagogy and cognitive process visualization, the model promotes the unified advancement of knowledge building and cognitive development, concurrently refining iterative teaching implementation pathways. Structured around the 5E instructional model and incorporating the orchestration of real-world scenarios—such as express delivery sorting during e-commerce festivals—the design integrates cognitive conflict induction and variational thinking training. An optimization process involving expert review and peer evaluation was conducted, clarifying the stage-specific functional positioning of the agents and effectively enhancing the operational feasibility and contextual relevance of the activities.

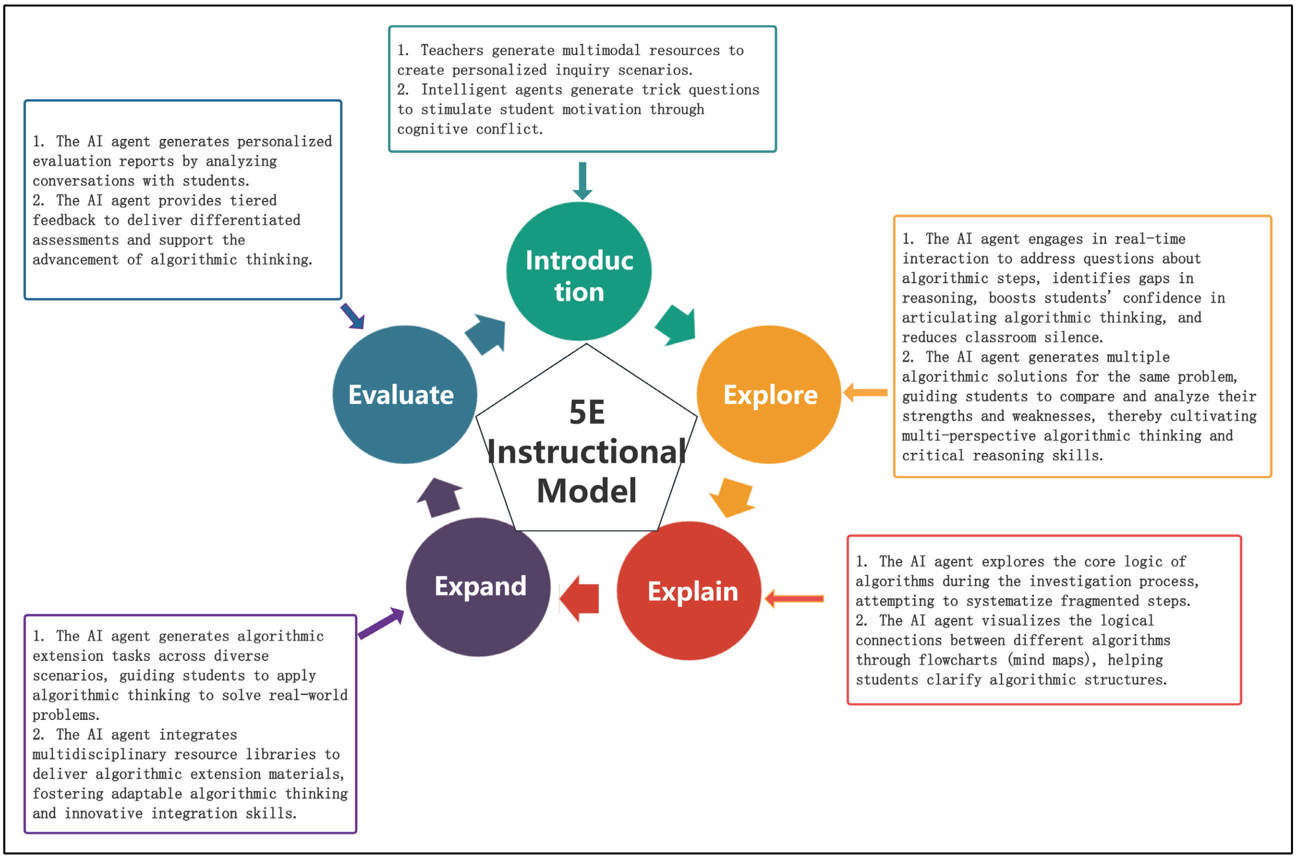

Figure 5. 5E teaching model integrating intelligent agents.

2) Based on the button platform, several teaching intelligent agents were developed to cultivate algorithmic thinking and serve different objectives

This study utilizes the low-code and modular features of ByteDance's Coze platform to develop an instructional agent, the "Intelligent Computing Partner," designed for the cultivation of algorithmic thinking among middle school students. The agent's design follows the cognitive developmental patterns of adolescents, emphasizing five constituent elements of algorithmic thinking: problem decomposition, pattern recognition, abstract modeling, algorithm design, and evaluation and optimization. Based on the "Thinking-Based Teaching Classroom" framework (Zhao et al., 2018), it establishes a personalized learning support system aligned with the developmental trajectory of algorithmic thinking. This system is operationalized through the multifaceted integration of four core agent roles: scenario creator, thinking guide, instant feedback provider, and learning partner. Functionally, it facilitates scenario visualization, provides tiered guidance, and delivers precise feedback, thereby effectively enhancing learning motivation and reinforcing the construction of independent thinking. In contrast to general-purpose generative AI tools, this agent reduces the learning threshold for students and supports teachers in optimizing instruction through recorded process data. The integration of educational theory and technological application embodied in this agent presents a targeted and practicable approach to fostering algorithmic thinking in secondary education.

6.2. Limitations and Future Prospects

The innovation of this study lies in transcending the educational limitations of general generative AI by developing a dedicated teaching agent and establishing a complete closed-loop process of "design, optimization, and evaluation." Its tripartite collaborative activity model and thinking-oriented design offer a novel paradigm for fostering core competencies in information technology education. This research, however, acknowledges certain limitations. The activity design primarily centers on sorting algorithms, and its applicability to other domains requires further exploration. Additionally, large-scale, longitudinal empirical validation has yet to be conducted, and the agent's adaptability for students at varying proficiency levels warrants more refined investigation.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

Adorni, G. (2023). *Dataset for Algorithmic Thinking Skills Assessment: Results from the Virtual CAT Pilot Study in Swiss Compulsory Education (Version 1.0.0) [Dataset]*. Zenodo.

Adorni, G., Piatti, S., & Karpenko, V. (2024). Virtual CAT: A Multi-Interface Educational Platform for Algorithmic Thinking Assessment. *SoftwareX, 27,* Article ID: 101737. https://doi.org/10.1016/j.softx.2024.101737

Bacelo, A., & Gómez-Chacón, I. M. (2023). Characterising Algorithmic Thinking: A University Study of Unplugged Activities. *Thinking Skills and Creativity, 48,* Article ID: 101284. https://doi.org/10.1016/j.tsc.2023.101284

Byrka, M. F., Sushchenko, A. V., Svatiev, A. V., Mazin, V. M., & Veritov, O. I. (2021). A New Dimension of Learning in Higher Education: Algorithmic Thinking. *Purposes Representations, 9,* e990.

Cui, J., Zhang, S. J., & Chen, J. (2025). Construction and Practice of AI-Enabled Programming Course Teaching Model. *Computer Technology and Development, 35,* 165-173. (In Chinese)

Huang, C. Q., Zhong, Y. H., Wang, X. Z. et al. (2025). From Single Agent to Multi-Agent: Design and Empirical Study of Incentive-Based Learning Activities Supported by Large-Model Agents. *Journal of East China Normal University (Educational Science Edition),* *43,* 44-56.

Ju, X. L., Zhang, Y. M., Wang, H. C. et al. (2024). Exploration of Python Programming Course Teaching Based on Large Language Model-Assisted Instruction. *Computer Education, No. 9,* 33-37. (In Chinese)

Lane, R. (2025). Mitigating Risks, Embracing Potential: A Framework for Integrating Generative Artificial Intelligence in Geographical and Environmental Education. *International Research in Geographical and Environmental Education,* 1-18. https://doi.org/10.1080/10382046.2025.2458561

Li, Z. L., Liu, C. H., Tao, Y. X., Shi, Y. N., & Li, S. J. (2020). Real-World Problem Solving: Teaching Strategies Aimed at Enhancing Core Competencies-A Case Study of the General Technology Curriculum at the High School Affiliated to Renmin University of China. *China Educational Technology, 2,* 109-116. (In Chinese)

Liao, J., Zhong, L., Zhe, L., Xu, H., Liu, M., & Xie, T. (2024). Scaffolding Computational Thinking with ChatGPT. *IEEE Transactions on Learning Technologies, 17,* 1628-1642. https://doi.org/10.1109/tlt.2024.3392896

Penney, J., Pimentel, J. F., Steinmacher, I., & Gerosa, M. A. (2024). Anticipating User Needs: Insights from Design Fiction on Conversational Agents for Computational Thinking. In A. Følstad, *et al.* (Eds.), *Chatbot Research and Design* (pp. 204-219). Springer. https://doi.org/10.1007/978-3-031-54975-5_12

Ritter, F., & Standl, B. (2023). Promoting Student Competencies in Informatics Education by Combining Semantic Waves and Algorithmic Thinking. *Informatics in Education, 22,* 141-160. https://doi.org/10.15388/infedu.2023.07

Sarı, U., Pektaş, H. M., Şen, Ö. F., & Çelik, H. (2022). Algorithmic Thinking Development through Physical Computing Activities with Arduino in STEM Education. *Education and Information Technologies, 27,* 6669-6689. https://doi.org/10.1007/s10639-022-10893-0

Sun, L., Zhu, Y. Q., Zhu, F., & He, W. S. (2024). Progressive Teaching Case Design and Practice for Cultivating Algorithmic Thinking Ability. *Computer Education, 11,* 182-186. (In Chinese)

Tang, L., Zhang, J. X., & Xu, Z. G. (2025). Application of Large Language Model in Computer Programming Practice Course Teaching. *Computer Education, No. 2,* 97-101+106. (In Chinese)

Tang, X. J., Lu, Y., Liu, N., Wang, H. W., & Gong, W. X. (2018). Research and Implementation of Online Open Course Quality Evaluation System Based on Delphi Method. *China Educational Informatization, 7,* 93-96. (In Chinese)

Tu, Z. Y., Zhou, K. J., Sun, M. T., Huang, Y., & Zhang, L. (2024). Opening the "Black Box" of Peer Review: An Analysis of the Behavioral Characteristics of Expert Reviewers. *Library Forum, 44,* 131-142. (In Chinese)

Wang, D. D., Zheng, Z. H., Zhou, L. J., & Fang, H. G. (2025). Research on Thinking Chain Teaching Design Based on Dialogic Teaching Theory and AI Agents. *Primary and Secondary School IT Education,* 32-35. https://kns.cnki.net/kcms2/article/abstract?v=k_902IEglk_s0F7zT9rc6f7wjtw4DtlF5oHkyFf-deqHqmsL7PzzL8L4rZT3DVMbfxKMIxL19mGfLvFYhuNmigu5ZLa-ZHzVYFs_swYMKTIj-tvLskXVoXgon9ss7jhqAqre-GJJAia6tPcb1yhcJYlWtbZFO0iLzVQG4eTXp54-M=&uniplatform=NZKPT

Wang, X., Xu, X., Zhang, Y., Hao, S., & Jie, W. (2024). Exploring the Impact of Artificial Intelligence Application in Personalized Learning Environments: Thematic Analysis of

Undergraduates' Perceptions in China. *Humanities and Social Sciences Communications, 11,* Article No. 1644. https://doi.org/10.1057/s41599-024-04168-x

Wu, X. X. (2024). A Brief Discussion on the "Context-Generation" Teaching Strategy in Junior High School Chinese Language Teaching. *Chinese Journal of Educational Studies, S2,* 52-53. (In Chinese)

Yu, Y., & Song, R. Y. (2024). Teaching Algorithmic Thinking in the Digital Age: Reverse Design Framework and Action Practice. *Journal of Distance Education, 42,* 86-94.

Zhai, X. S., Zhang, L. J., Xia, L. L. et al. (2024). Research on the Application of GAI-Based Reverse Engineering Teaching Thinking in Human-Computer Collaboration-Taking Programming Education as an Example. *Research on Educational Technology, 45,* 61-68. (In Chinese)

Zhao, G. Q., Xiong, Y. W., & Wang, X. L. (2018). The Concept, Elements and Design of Thinking Development Classroom. *China Educational Technology, No. 7,* 7-15. (In Chinese)