

ประสิทธิผลของการจัดการเรียนรู้การเขียนโปรแกรม C++ โดยใช้ปัญญาประดิษฐ์ช่วยในระดับประถมศึกษา: แนวทางการวิเคราะห์การเรียนรู้

Qiu Weiyang*, สุดาสวรรค์ งามมงคลวงศ์

คณะเทคโนโลยีดิจิทัลและนวัตกรรม

มหาวิทยาลัยเซาธ์อีสต์บางกอก กรุงเทพมหานคร

*Corresponding author email: kewyang55@gmail.com

รับบทความ: 23 เมษายน 2569, รับบทความแก้ไข: 31 พฤษภาคม 2569, ยอมรับตีพิมพ์: 28 มิถุนายน 2569

บทคัดย่อ

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อศึกษาประสิทธิผลของรูปแบบการจัดการเรียนการสอนที่ใช้ปัญญาประดิษฐ์ (AI) เป็นเครื่องมือสนับสนุนร่วมกับการชี้แนะของครูต่อผลสัมฤทธิ์ทางการเรียนด้านการเขียนโปรแกรมภาษา C++ ของนักเรียนระดับประถมศึกษา และเพื่อวิเคราะห์ความสัมพันธ์ระหว่างพฤติกรรมการเรียนรู้กับผลสัมฤทธิ์ทางการเรียน กลุ่มตัวอย่างประกอบด้วยนักเรียนชั้นประถมศึกษาปีที่ 5 จากโรงเรียน Hanlin Experimental School จำนวน 78 คน แบ่งเป็นกลุ่มทดลอง 39 คน และกลุ่มควบคุม 39 คน เครื่องมือที่ใช้ในการวิจัย ได้แก่ แบบทดสอบวัดผลสัมฤทธิ์ทางการเรียน และระบบวิเคราะห์ข้อมูลการเรียนรู้ การวิเคราะห์ข้อมูลใช้ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐาน การทดสอบค่าทีสำหรับกลุ่มอิสระ (Independent Samples t-test) และค่าสัมประสิทธิ์สหสัมพันธ์ของเพียร์สัน (Pearson's Correlation Coefficient)

ผลการวิจัยพบว่า กลุ่มทดลองมีคะแนนหลังเรียนสูงกว่ากลุ่มควบคุมอย่างมีนัยสำคัญทางสถิติ โดยมีคะแนนเฉลี่ยเท่ากับ ($M = 61.38$, $SD = 5.66$) ขณะที่กลุ่มควบคุมมีคะแนนเฉลี่ยเท่ากับ ($M = 53.41$, $SD = 7.24$) และมีขนาดอิทธิพลอยู่ในระดับสูง ($d=1.23$) นอกจากนี้ ความถูกต้องของการส่งงานมีความสัมพันธ์เชิงบวกกับผลสัมฤทธิ์ทางการเรียนอย่างมีนัยสำคัญทางสถิติ ($r = .339$, $p < .05$) ในขณะที่ปริมาณการฝึกปฏิบัติไม่พบความสัมพันธ์อย่างมีนัยสำคัญทางสถิติ ($r = .034$, $p > .05$)

ผลการศึกษาชี้ให้เห็นว่า ความสำเร็จในการเรียนรู้การเขียนโปรแกรมขึ้นอยู่กับคุณภาพของการมีส่วนร่วมในกระบวนการเรียนรู้มากกว่าปริมาณการฝึกปฏิบัติเพียงอย่างเดียว สำหรับผู้สอน ผลการวิจัยนี้สะท้อนให้เห็นถึงความสำคัญของการออกแบบกลยุทธ์การสอนที่มุ่งเน้นกระบวนการเรียนรู้ การให้คำแนะนำในการแก้ไขข้อผิดพลาดของโปรแกรมอย่าง

เป็นระบบ มากกว่าการเน้นการฝึกซ้ำในปริมาณมากเพียงอย่างเดียว นอกจากนี้ งานวิจัยยังมี ส่วนช่วยขยายองค์ความรู้ด้านการจัดการศึกษาที่ใช้ปัญญาประดิษฐ์เป็นเครื่องมือสนับสนุน โดยแสดงให้เห็นว่าการบูรณาการการสนับสนุนการเรียนรู้ด้วย AI ควบคู่กับการชี้แนะของครู สามารถช่วยลดภาระทางการรับรู้ (Cognitive Load) และส่งเสริมการพัฒนาทักษะการคิดเชิง คำนวณ (Computational Thinking) ของผู้เรียนที่เริ่มต้นเรียนรู้การเขียนโปรแกรมได้อย่างมีประสิทธิภาพ

คำสำคัญ: การจัดการเรียนรู้การเขียนโปรแกรม/ การวิเคราะห์ข้อมูลการเรียนรู้/
การเรียนรู้โดยใช้ปัญญาประดิษฐ์/ การคิดเชิงคำนวณ/ ระดับประถมศึกษา

Effectiveness of AI-Assisted C++ Programming Instruction in Primary Education: A Learning Analytics Approach

Qiu Weiyang*, Sudasawan Ngammongkolwong

Faculty of Digital Technology and Innovation,
Southeast Bangkok University, Bangkok

*Corresponding author email: kewyang55@gmail.com

Received: 23 April 2026, Revised: 31 May 2026, Accepted: 28 June 2026

Abstract

The purpose of this study was to examine the effectiveness of an AI-assisted and teacher-guided instructional model on primary school students' C++ programming achievement and to analyze the relationship between learning behaviors and academic outcomes. The sample consisted of 78 fifth-grade students from Hanlin Experimental School, divided into an experimental group ($n=39$) and a control group ($n=39$). The research instruments included an achievement test and a learning analytics system. Data were analyzed using mean, standard deviation, independent samples t-test, and Pearson's correlation coefficient.

The results revealed that the experimental group had significantly higher post-test scores ($M = 61.38$, $SD = 5.66$) than the control group ($M = 53.41$, $SD = 7.24$) with a large effect size ($d = 1.23$). Additionally, submission accuracy showed a significant positive correlation with learning achievement ($r = .339, p < .05$) whereas practice volume showed no significant correlation ($r = .034, p > .05$).

The findings suggest that effective programming achievement depends more on the quality of engagement than the quantity of practice. For educators, this highlights the practical implication of prioritizing process-oriented pedagogical strategies and guided debugging over mere quantity-based repetition. Furthermore, this study contributes to AI-assisted education research by demonstrating that integrating AI scaffolding with teacher guidance

effectively mitigates cognitive load and fosters computational thinking in novice programmers.

Keywords: Programming Education/ Learning Analytics/ AI-Assisted Learning/
Computational Thinking/ Primary Education

Introduction

With the rapid advancement of artificial intelligence (AI) and digital technologies, programming competence has increasingly been recognized as a core skill for learners in the 21st century. In response to this global trend, the Organisation for Economic Co-operation and Development (OECD) incorporated computational thinking into the Programme for International Student Assessment (PISA) framework for the first time in 2022, highlighting its importance as a foundational competency for future-oriented education. [1] Similarly, global educational reforms have prioritized computer science education. A comparative analysis regarding K-12 computer science curricula in ten countries confirms that integrating programming as a compulsory subject has become a dominant trend since 2014 [2]. Aligning with this trend, China's "New Generation Artificial Intelligence Development Plan" explicitly proposes the gradual implementation of programming education within compulsory education [3], followed by the official "Information Science and Technology Curriculum Standards (2022 Edition)" [4]. According to these standards, primary school students are required to master at least one visual programming tool, while students in higher grades are encouraged to explore text-based programming languages such as C++.

While visual programming languages like Scratch lower entry barriers, Weintrop and Wilensky's (2017) research on the 'modality effect' argues that transitioning to text-based languages (e.g., C++) is essential for students to develop deeper conceptual understanding and professional coding practices. Despite these advancements, elementary-level C++ research is still significantly limited. This is primarily because the high syntactic complexity of text-based languages presents a substantial cognitive barrier, making it a subject typically reserved for older students [5]. Furthermore, previous studies have largely failed to examine the micro-level learning processes of these young novices, relying instead on summative assessments that only capture final outcomes. As Grover and Pea [6] highlight, assessing computational thinking is complex; relying solely on final products fails to capture the 'process data' of how students debug and

refine their logic. Consequently, valuable learning behaviors that occur during the programming process remain underexplored.

Recent studies in the Chinese context indicate that current programming education in primary schools faces challenges such as weak student foundations despite high interest [7]. To address these challenges, modern programming education aims to bridge the gap between abstract algorithmic concepts and concrete real-world applications. Beyond the classic definition, recent research emphasizes the "dynamic interplay" between CT practices and self-regulated learning [8]. Learning Analytics (LA) shifts assessment from "result-based" to "process-based". Recent frameworks utilize multimodal data—such as compilation logs, error frequencies, and time-on-task—to predict student performance and identify "at-risk" students early [9].

Constructivism posits that learners actively build knowledge based on experience, and in C++ programming, students often face a high cognitive load. To overcome this, AI-assisted scaffolding is specifically needed for novice learners. Unguided exposure to C++ syntax often leads to overwhelming frustration and task abandonment among primary students.

AI-assisted learning involves using AI technologies to provide personalized support. Hwang et al.[10] highlight that the role of AI in education has evolved from simple automated grading to acting as an "intelligent partner" that offers diagnostic feedback and personalized guidance [10]. Immediate AI feedback acts as a necessary 'intellectual safety net,' automatically resolving low-level syntax errors so that young learners can focus on higher-order algorithmic thinking. The necessity of implementing "scaffolding" strategies effectively mitigates cognitive overload and supports young learners' transition to complex syntax (Swidan et al., 2018).

Against this backdrop, the present study aims to examine the learning achievement and behavioral processes of primary school students in C++ programming. Specifically, this study addresses three research questions:

1. Does AI-assisted instruction improve primary students' C++ achievement?

2. What learning behaviors emerge during AI-assisted programming learning?

3. Which learning behaviors predict programming achievement?

By answering these questions, this research provides frontline teachers with a feasible framework for collecting and analyzing learning data, facilitating a shift from experience-based teaching to evidence-based teaching.

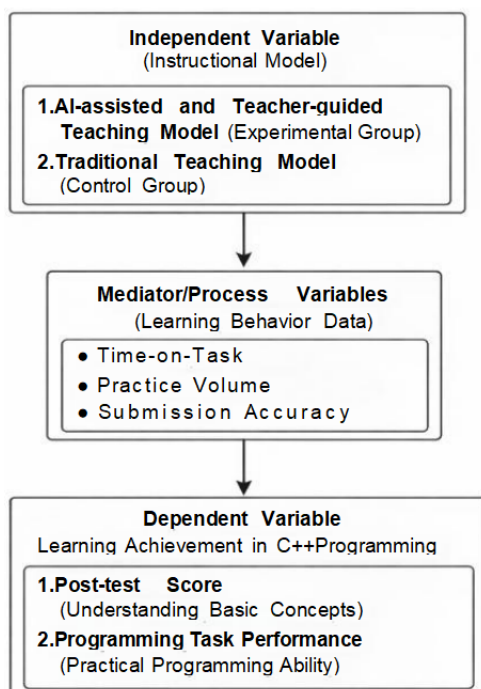


Figure 1 Conceptual Framework of the Study

Materials and Methods

This study employed a quantitative research approach using a quasi-experimental research design, specifically the pretest–posttest control group design. This design was selected to examine the effectiveness of different instructional approaches in primary school C++ programming education while controlling for initial differences in students’ prior knowledge.

Population and Sample: The sample comprised 78 fifth-grade students, selected using purposive sampling based on the criteria of being enrolled in

Grade 5 (aged between 11 and 12 years) with no prior systematic learning experience in C++ programming. The participants were divided into two groups: an experimental group (N=39) receiving instruction through an AI-assisted and teacher-guided teaching model, and a control group (N=39) receiving instruction through a traditional teaching approach without AI assistance. The instructional experiment was conducted covering a total of 12 weeks with two instructional hours per week.

Research Instruments and Data Collection: Three main research instruments were used in this study.

1. Achievement Test: Developed to assess students' learning achievement in fundamental C++ programming concepts (Sequence, Selection, Loop, Array, and Function). It was administered as both a pretest and a posttest.

2. AI-Assisted and Teacher-Guided Instruction: The experimental group utilized the 'AI Link Science' platform, a web-based environment integrating a C++ compiler with an AI tutoring system and a backend data logging module. Specifically, the AI component functioned as an on-demand tutor, providing automated error diagnosis, real-time coding hints, and an embedded chatbot that offered immediate explanations to resolve students' confusion regarding code logic or syntax. Within this blended learning process, the explicit role of the instructor was to provide high-level guidance on computational thinking and problem decomposition, while actively supervising the AI's real-time feedback. This collaborative model allowed the AI to manage repetitive technical scaffolding, enabling the teacher to focus on deeper pedagogical interventions.

3. Learning Behavior Indicators: Three key quantitative indicators were automatically collected to measure students' learning behaviors: Time-on-Task (average duration solving problems), Practice Volume (total number of attempted exercises), and Submission Accuracy Rate (percentage of passed test cases).

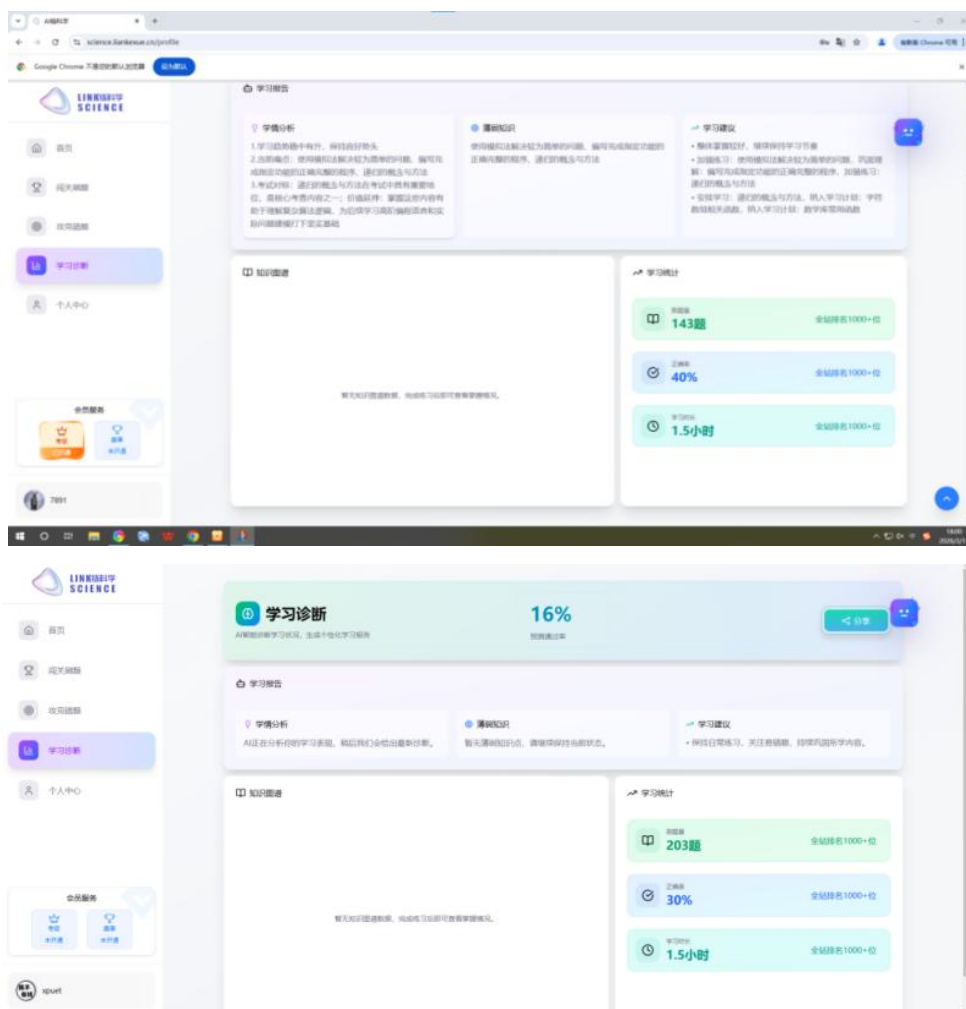


Figure 2 User Interface and Learning Report of the AI Link Science Platform



求三个整数最大值的C++代码分析与优化

输出格式
排序后的3个数。

输入/输出例子1
输入：
5 10 6
输出：
10 6 5

如何用C++解决

解题思路
我们需要输入3个整数，从大到小排序后输出。给你两种最简单、最适合新手的写法：

方法1：手动比较排序（适合理解逻辑）
直接通过判断交换数值，把最大的放前面，中间的放中间，最小的放后面。

```
cpp
#include <iostream>
using namespace std;

int main() {
    int a, b, c, temp;
    cin >> a >> b >> c;

    // 第一步：让 a 变成最大的数
    if (b > a) {
        temp = a;
        a = b;
        b = temp;
    }
}
```

发消息或输入“/”选择技能

在计算机中，每个二进制位都可以表示为 0 或 1 两个值。
()

对
错

解析：
二进制位 (bit) 是计算机中最小的数据单位，只能表示 0 或 1 两种状态。

因为二进制的定义就是逢二进一，它只有两个基本数字：0 和 1。

- 计算机底层是用电路的通 / 断、电压的高 / 低来存储和处理数据的
- 电路只有两种稳定状态，刚好对应 0 和 1
- 所以一个二进制位 (bit) 只能表示 0 或 1，不可能有第三个值

因此这句话是对的。

为什么



求三个整数最大值的C++代码分析与优化

输出格式
排序后的3个数。

输入/输出例子1
输入：
5 10 6
输出：
10 6 5

如何用C++解决

解题思路
我们需要输入3个整数，从大到小排序后输出。给你两种最简单、最适合新手的写法：

方法1：手动比较排序（适合理解逻辑）
直接通过判断交换数值，把最大的放前面，中间的放中间，最小的放后面。

```
cpp
#include <iostream>
using namespace std;

int main() {
    int a, b, c, temp;
    cin >> a >> b >> c;

    // 第一步：让 a 变成最大的数
    if (b > a) {
        temp = a;
        a = b;
        b = temp;
    }
}
```

发消息或输入“/”选择技能

Figure 3 Examples of AI-Assisted Tutoring and Student-AI Interaction

The instruments were reviewed by three experts in education and information technology, yielding an Index of Item-Objective Congruence (IOC) between 0.86 and 1.00, ensuring high content validity. Additionally, the reliability of the achievement test was examined using Cronbach's alpha coefficient, with a value of 0.70 or higher considered acceptable, indicating adequate internal consistency of the instrument.

Data were analyzed using descriptive statistics (Mean, Standard Deviation) and inferential statistics (Independent samples t-test and Pearson's correlation coefficient) with a significance level set at $\alpha = 0.05$.

Ethical Considerations: Prior to the commencement of the experiment, informed consent was obtained from all participating students, and explicit written permission was secured from their parents or legal guardians. The study strictly adhered to ethical guidelines regarding student privacy protection. All learning behavior data collected via the platform were fully anonymized, stored securely, and utilized exclusively for academic research purposes to ensure complete confidentiality.

Results

Analysis of Learning Achievement (RQ1)

To evaluate the effectiveness of the proposed instructional model, Independent Samples t-tests were conducted on both pre-test and post-test scores to compare the Experimental Group and the Control Group.

Pre-test Analysis (Baseline Check)

Prior to the intervention, a pre-test was administered to both groups (N=78) to assess their prior knowledge of C++ programming.

Table 1 Independent Samples t-test Results for Pre-test and Post-test Scores

Period	Group	N	Mean	Std. Deviation	t	df	p	Cohen's d
Pre_Test	Control	39	49.21	8.345	1.632	76	0.107	-0.37
	Experimental	39	46.23	7.737				
Post_Test	Control	39	53.41	7.236	-5.421	76	< .001	1.23
	Experimental	39	61.38	5.66				

As shown in Table 1, the mean pre-test score for the Control Group was 49.21 (SD = 8.35), while the Experimental Group was 46.23 (SD = 7.74). The independent samples t-test results indicated no statistically significant difference between the two groups ($t = 1.632, df = 76, p = .107 > .05$).

To further assess the baseline difference, Cohen's d was calculated. The resulting effect size was $d = -0.37$, which is considered a small effect size. This confirms that students in both groups had equivalent starting levels of programming competency, satisfying the baseline requirement for a quasi-experiment.

Post-test Analysis (Intervention Effect)

After the 12-week instructional period, a post-test was conducted to measure learning achievement.

Table 1 reveals the results of the post-test. The experimental group demonstrated a statistically significant improvement in post-test performance ($M=61.38, SD=5.66$), yielding a relatively higher level of achievement compared to the control group. The t-test analysis demonstrates a statistically significant difference ($t = -5.421, p < .001$).

Moreover, the effect size Cohen's d was calculated to be 1.23. According to Cohen's convention ($d > 0.8$ is large), this indicates a very large effect size. This finding strongly supports the hypothesis that the AI-assisted instructional model not only produces a statistically significant improvement but also has a substantial practical impact on students' learning achievement compared to the traditional teaching model.

Analysis of Learning Behaviors (RQ2)

Unlike the traditional classroom where learning processes are difficult to quantify, the "AI Link Science" platform automatically logged granular learning data for the Experimental Group. Therefore, this section specifically focuses on analyzing the behavioral characteristics of students in the experimental group

(N=39). Data for the Control Group were not included in this analysis due to the lack of automated logging tools in the traditional instruction environment.

Table 2 Descriptive Statistics of learning behaviors

	N	Minimum	Maximum	Mean	Std. Deviation
Practice_Volume	39	32	67	47.05	7.067
Submission_Accuracy	39	0.659	0.88	0.75826	0.05544
Time_on_Task_Min	39	3	13.3	8.908	2.3773
Valid N (listwise)	39				

As presented in Table 2, the students in the experimental group demonstrated the following behavioral patterns:

1.Practice Volume: Students completed an average of 47.05 exercises (SD = 7.07). This indicates a high and sustained level of engagement, with students completing approximately 4 exercises per week on average.

2.Submission Accuracy: The average submission accuracy was 0.76 (75.83%). This reflects a realistic learning curve where students successfully resolved three out of four submission errors with AI assistance.

3.Time-on-Task: The average time spent per task was 8.91 minutes(SD = 2.38), suggesting that students were engaged in deep processing and debugging rather than rapid guessing.

Table 3 Correlations Matrix

		Post_Test	Practice_V olume	Submission_ Accuracy	Time_on_Task _Min
Post_Test	Pearson	1	.034	.339*	-.036
	Correlation				
	Sig. (2-tailed)		.835	.035	.828
	N	78	39	39	39
Practice_Volume	Pearson	.034	1	.084	-.342*
	Correlation				
	Sig. (2-tailed)	.835		.613	.033
	N	39	39	39	39
Submission_Accuracy	Pearson	.339*	.084	1	.209
	Correlation				
	Sig. (2-tailed)	.035	.613		.201
	N	39	39	39	39
Time_on_Task_Min	Pearson	-.036	-.342*	.209	1
	Correlation				
	Sig. (2-tailed)	.828	.033	.201	
	N	39	39	39	39

*. Correlation is significant at the 0.05 level (2-tailed).

Correlation Analysis (RQ3)

To investigate whether these learning behaviors predicted final learning achievement, a Pearson product-moment correlation analysis was conducted for the Experimental Group. The results are presented in Table 3.

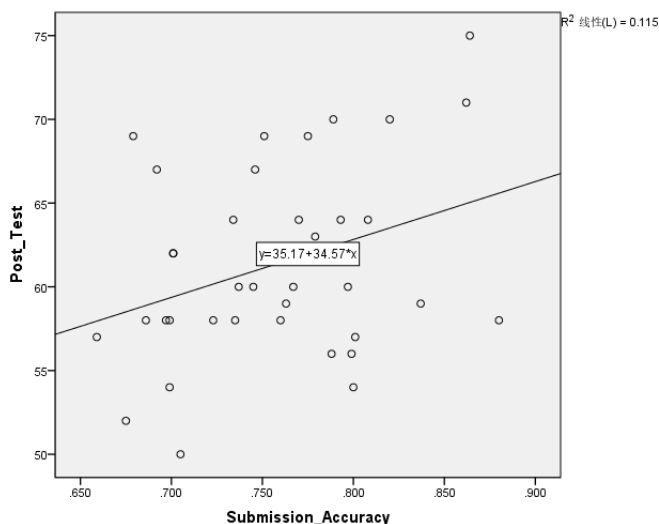


Figure 4 Scatter Plot of Submission Accuracy vs. Post-Test Score

2.Practice Volume and Achievement ($r = .034, p = .835 > .05$):

The correlation between Practice Volume and Post-test Scores was not statistically significant.

3.Practice Volume and Time-on-Task ($r = -.342 *, p = .033 < .05$):

A significant negative correlation was found here.

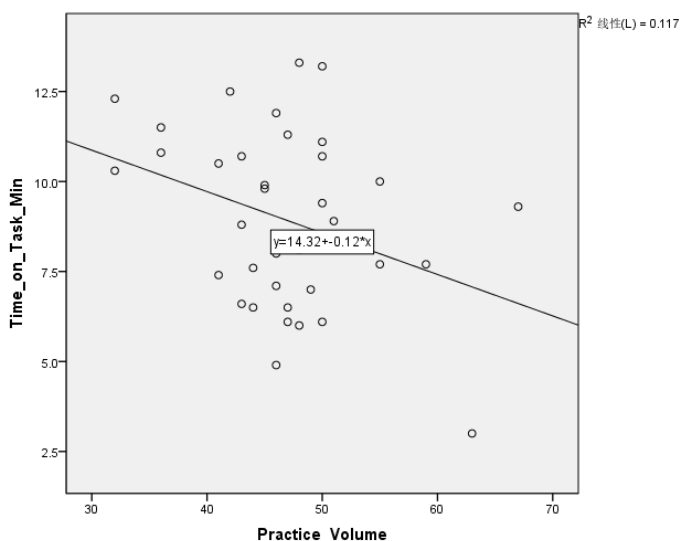


Figure 5 Scatter Plot of Practice Volume vs. Time-on-Task

Discussion

The Effectiveness of AI Scaffolding:

The highly significant improvement in post-test scores ($p < .001$) aligns with Hwang et al. [10], confirming that AI-assisted instruction outperforms traditional methods. The Control Group's lower mean reflects the high cognitive load of text-based programming described by Weintrop and Wilensky(2017). Without immediate feedback, traditional students likely struggled with syntax errors. In contrast, the AI platform acted as an "intellectual safety net." The relatively high average time-on-task combined with high practice volume suggests that the AI's real-time hints sustained student engagement, enabling them to persist through complex C++ logic. The system's ability to automatically resolve low-level syntax issues through error diagnosis freed up cognitive resources, allowing students to focus on algorithmic problem-solving.

Quality Over Quantity in Programming Practice:

The findings from the correlation analysis provide a nuanced insight that can be deeply explained through the lens of self-regulated learning (SRL) theory. In the context of SRL, practice volume typically serves as a primary indicator of learning engagement and diligence. However, contrary to the common belief that 'practice makes perfect,' our data shows that Practice Volume was not significantly correlated with final scores, while Submission Accuracy was a significant predictor.

This finding underscores the dynamic interplay between computational thinking and self-regulated learning processes. It suggests that mere repetition—reflecting a low level of self-monitoring—is ineffective. High-performing students did not merely increase their task volume; they exhibited advanced self-regulation by engaging in 'deliberate practice'. By actively monitoring their submission accuracy, evaluating AI-generated feedback, and reflecting on logical correctness, these students successfully engaged in reflective problem-solving.

This supports the premise that in an AI-assisted environment, true academic success relies on learners exercising high-quality self-regulation to transform external AI scaffolding into internalized cognitive growth.

Efficiency Gains:

The significant negative correlation between Practice Volume and Time-on-Task indicates a "proficiency effect." As students solved more problems, they became faster at coding and debugging, which improved procedural fluency. However, since volume didn't correlate with the final score, teachers should emphasize that speed should not come at the expense of understanding. The AI platform transformed "blind guessing" into "informed debugging," where the AI provides the necessary scaffolding to help students bridge their Zone of Proximal Development independently.

Critical Reflections and Potential Challenges :

While the AI-assisted model demonstrates significant pedagogical benefits, it is crucial to critically acknowledge its potential negative impacts and the inherent limitations of AI-generated feedback. A primary concern is the risk of student over-reliance on AI scaffolding. If learners continuously rely on AI to automatically resolve errors without engaging in deep cognitive reflection, this dependency may hinder the development of their independent problem-solving capabilities. Furthermore, the effectiveness of the AI intervention may vary significantly among different types of learners. Highly self-regulated students tend to utilize AI hints constructively for deliberate practice, whereas students with lower self-regulation or weaker foundational knowledge might struggle to interpret complex AI explanations or simply "game the system" for quick answers. Lastly, current AI-generated feedback has technical limitations; it may occasionally misinterpret a student's unique logical intent or provide generalized diagnostic hints that fail to address the specific root of a misconception. Therefore, teachers must continuously monitor AI-student

interactions to prevent over-reliance and provide differentiated support for diverse learner profiles.

Conclusion

This study confirms that an AI-assisted instructional model significantly enhances fifth-grade students' C++ programming achievement compared to traditional methods. Within the AI environment, the quality of learner engagement—specifically submission accuracy—proved to be a far more critical predictor of academic success than the sheer volume of practice.

Beyond classroom practice, these findings carry broader educational implications. At the curriculum policy level, policymakers should consider formalizing the integration of AI-assisted frameworks into K-12 education standards, which could safely lower the entry barrier for complex text-based languages like C++. Furthermore, the successful implementation of such models necessitates targeted teacher professional development. Training programs must equip educators with the skills to interpret learning analytics effectively, empowering them to shift their instructional roles from basic syntax correctors to high-level facilitators of computational thinking. Institutions are therefore encouraged to adopt blended "Teacher-Led + AI-Assisted" models to maximize pedagogical efficiency.

Despite the positive outcomes, this study has certain limitations that must be acknowledged. First, the sample size was relatively small ($n=78$) and restricted to a single primary school setting, which may limit the generalizability of the findings to broader demographic contexts. Second, the 12-week intervention period was relatively short. Future studies should aim to involve larger, multi-school cohorts over an extended timeframe to evaluate long-term skill retention. Additionally, advanced causal modeling (e.g., Structural Equation Modeling) should be employed to rigorously examine the causal pathways of the "AI $\rightarrow$ Behavior $\rightarrow$ Achievement" model.

References

1. OECD. PISA 2022 assessment and analytical framework. Paris: OECD Publishing; 2023 Aug 31. doi: 10.1787/dfe0bf9c-en
2. Heintz F, Mannila L, Färnqvist T. A review of models for introducing computational thinking, computer science and computing in K-12 education. In: 2016 IEEE Frontiers in Education Conference (FIE); 2016 Oct 12-15; Erie, PA, USA. IEEE; 2016. p. 1-9. doi: 10.1109/FIE.2016.7757410.
3. State Council of the People's Republic of China. New generation artificial intelligence development plan (State Council Document No. 35). Beijing: State Council; 2017.
4. Ministry of Education Information science and technology curriculum standards for compulsory education (2022 edition). Beijing: Beijing Normal University Publishing Group; 2022.
5. Swidan A, Hermans F, Smit M. Programming with scratch or python? The effect of programming environments on novice programmers' cognitive load. In: 2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET); 2018 May 27-Jun 3; Gothenburg, Sweden. IEEE; 2018. p. 57-66. doi: 10.1145/3183377.3183393
6. Grover S, Pea R. Computational thinking in K-12: A review of the state of the field. Educ Res. 2013;42(1):38-43. doi: 10.3102/ 0013189X12463051
7. Ou Y. Investigation and analysis of the current situation of programming education in primary and secondary schools. Heliyon. 2023;9(4):e15530. doi: 10.1016/j.heliyon.2023.e15530
8. Kong S-C, Wang YQ. Dynamic interplays between self-regulated learning and computational thinking in primary school students through animations and worksheets. Comput Educ. 2024;219: Article 105096. doi: 10.1016/j.compedu.2024.105126

9. Qiu F, Zhang G, Sheng X, Jiang L, Zhu L, Xiang Q, et al. Predicting students' performance in e-learning using learning process and behaviour data. *Sci Rep.* 2022;12(1): Article 453. doi: 10.1038/s41598-021-03867-8
10. Hwang G-J, Xie H, Wah BW, Gašević D. Vision, challenges, roles and research issues of artificial intelligence in education. *Comput Educ: Artif Intell.* 2020;1: Article 100001. doi: 10.1016/j.caeai.2020.100001