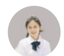
How AI assisted K-12 Computer Science Education? A Systematic Review

Conference Paper · June 2024

DOI: 10.18260/1-2--47532

Zifeng Liu
University of Florida



Xinyue Jiao
New York University

Xueyan Gao
University of Florida

Hyunju Oh
University of Florida

How AI assisted K-12 Computer Science Education? A Systematic Review

Abstract

Although computational thinking is critical in education, not only to enhance students' problem-solving and logical thinking skills but also to broaden their creativity and understanding of systems design, challenges such as inadequate educational resources, lack of teaching experience, and abstract nature of programming principles continue to hinder the promotion and implementation of high-quality computer science (CS) education. Artificial intelligence (AI) holds promise in addressing these issues. Yet, the specific impact of AI on K-12 CS education has to be discussed. Existing reviews have focused on the broad spectrum of AI applications in education, with relatively little focus on topics related to CS education and programming instruction, with most of these studies focusing on a single type of AI, such as automated evaluation systems or visual programming, and failing to fully cover the various categories of AI, including machine learning, deep learning, and robotics, especially in the K-12 field. The primary goal of this study is to conduct a systematic review of the current literature concerning the role, impact, and constraints of AI in CS education, with a specific focus on K-12 education. The review process follows the PRISMA principle. A total of 24 articles published between 2013 and 2023 were selected, comprehensively reviewed, and analyzed. The coding scheme mainly includes four aspects: (1) Research background, (2) Research design, (3) AI technologies, and (4) Research outcomes and limitations. Each aspect contains specific dimensions to be coded. The study discovered that AI plays a significant role in K-12 CS as learning content and developing programming platforms. These adaptive learning platforms give personalized programming education and real-time feedback, relieving teachers' workload while giving students personalized curricular information tailored to their needs. Additionally, AI is usually used as a data analytics tool to predict student performance. The reviewed articles focus on AI's cognitive and affective impact on students and found positive effects on those variables. At the same time, AI allows for better analysis and utilization of data on student behavior while programming. Limitations in the current reviewed articles on AI in K-12 CS education include insufficient attention to theoretical adoption, ethical concerns, and methodological issues like small sample sizes. This review highlights the critical role of AI in K-12 CS education and illuminates directions for a more personalized, interactive, and practical learning experience in K-12 CS education in the future.

Introduction

As technology becomes increasingly important in our society, it's crucial to equip the new generation of K-12 students with computational thinking skills[1,2]. Computational thinking goes beyond programming abilities; it encompasses problem-solving approaches, data analysis, and system design[3]. Given its significance, computer science (CS) education has gained increasing attention as the curriculum for nurturing students' computational thinking abilities[4,5]. However, there are multiple challenges faced with CS education at K-12 level. Firstly, the abstract nature of programming principles, along with complicated algorithm designs, can be intimidating to students, thereby dampening their interest in computational thinking[6]. Furthermore, inconsistent standards for varying levels of teachers training and CS courses across regions make it difficult to ensure that all K-12 students receive high-quality CS education[7].

Artificial intelligence (AI) has the potential to offer personalized content[8] and feedback[9], thereby making high-quality CS education accessible to a broader range of students[10,11]. Besides, by incorporating enjoyable interaction applications and games, AI can make the learning process more appealing to students, lower the difficulty of the process, and enhance their interest in learning[12,10]. Furthermore, AI can automate teaching and evaluating processes and provide training and resources for CS teachers to improve their education, thus providing students equal access to quality CS education[13,14].The CSTA K-12 CS standards[15] provide a comprehensive framework essential for integrating AI into K-12 CS education. The standards emphasize not only technical proficiency in CS but also critical thinking and problem-solving skills, preparing students to navigate and contribute to an AI-driven future.

Most of the existing review articles have primarily focused on the broad spectrum of AI applications within the realm of education[16,17,18,19]. Some of these reviews have extended their focus towards specialized categories of AI applications in education, such as robotics[20], feedback systems[21,22], and intelligent tutoring systems[23]. Certain studies have delved into the analysis of nurturing AI literacy within the K-12 educational domain[24], as well as the ethical challenges confronting the application of AI in educational settings[25]. Nonetheless, there still lacks a systematic review regarding AI's roles and effects on CS education, especially for K-12 education. Thus, we formulated our research question as below:

RQ1: What are the publication and study characteristics of K-12 CS Education with AI?

RQ2: How does AI impact student learning in K-12 CS education?

RQ3: What are the most used AI-assisted strategies and tools for teaching CS in K-12 education?

Methods

Search strategy and selection procedure

Keywords and search strings as in Table 1 were used to search different databases including ProQuest (including ERIC), Scopus, Web of Science, IEEE Xplore, and ACM Digital Library, with the included years limited to 2013 to 2023. We choose this time frame because it represents

Table 1: Search Strings

Topic	Operator	Search string elements
AI	AND	Artificial Intelligence or AI or Machine Learning or ML or Deep Learning or DL or Data Mining or DM or Natural Language Process* or NLP or Computer Vision or CV or Robot* or Intelligent*
K-12	AND	k-12 or elementary school or primary school or middle school or high school or secondary school
CS Education	AND	computer science or cs or programming or cod* or comput*

a significant period in the advancement and integration of AI technologies in education, as shown in previous studies[4,26,27]. The literature obtained from the search was then subjected to an initial screening. The literature screening process followed the PRISMA guidelines[28].

Inclusion and Exclusion Criteria

This study conducted an initial screening with 2661 selected articles. The selection criteria for the initial screening was that the paper was an English journal article, and 1509 documents related to the research topic were selected according to the Figure 1 as shown in the figure. These articles were then reviewed and analyzed in detail to identify articles focusing on the impact and application of AI in K-12 CS education and to narrow the number of articles by taking into account the impact factor of the journals in which the selected articles were published. Finally, 24 studies were included for further analyze.

Data extraction, analysis, and synthesis

This paper primarily analyzed four aspects of the selected literature (shown in Table 2). It comprises several elements each with different dimensions and types to classify the literature's characteristics. For instance, under "Research background," the country or region dimension categorizes studies by geographical origin. Educational level further refines the classification, allowing for a comparison of research across different school levels or the absence of such specification. Note: In Table 2, several acronyms are used to denote various educational technologies. 'ITS' stands for Intelligent Tutoring Systems, 'AAS' refers to Automatic Assessment Systems, and 'PAT' denotes Programming Assistance Tools. 'VLS' represents Virtual Labs and Simulations, while 'LA' signifies Learning Analytics. The acronym 'NLP' is used for Natural Language Processing Tools, and 'CG' for Computer Programming Educational Games. Additionally, 'ML&DL Education' refers to Machine Learning and Deep Learning Education, and 'CV&SR' stands for Computer Vision and Speech Recognition Tools.

Table 2: The coding scheme.

Element	Dimension	Type
Research Background	Country/Region	1.USA 2.Asia 3.Europe 4.Others
	Educational Level	1. Primary school 2. Middle school 3. High school 4. Not mention
	Theoretical Framework Adopted	1. Yes 2. No
Research Design	Research method	1. Quantitative methods 2. Qualitative methods 3. Mixed methods
	Data/Sample Size	1. Small scale (<50) 2. Medium scale (51–300) 3. Large scale (>300)
	Dependent Variable	1. Learning outcome 2. Engagement 3. User experience 4. Model accuracy 5. Others
	RQ/Goals	/
AI Technologies	AI types	1. ML 2. DL 3. NLP 4. CV 5. Robotics
	AI roles	1. ITS 2. AAS 3. PAT 4. VLS 5. LA 6. NLP 7. CG 8. ML & DL Education 9. CV & SR
	Effective Evaluation	1. Positive 2. Negative
Research Outcome	Ethical & Fairness Considerations	1. Yes 2. No
	Limitations	/

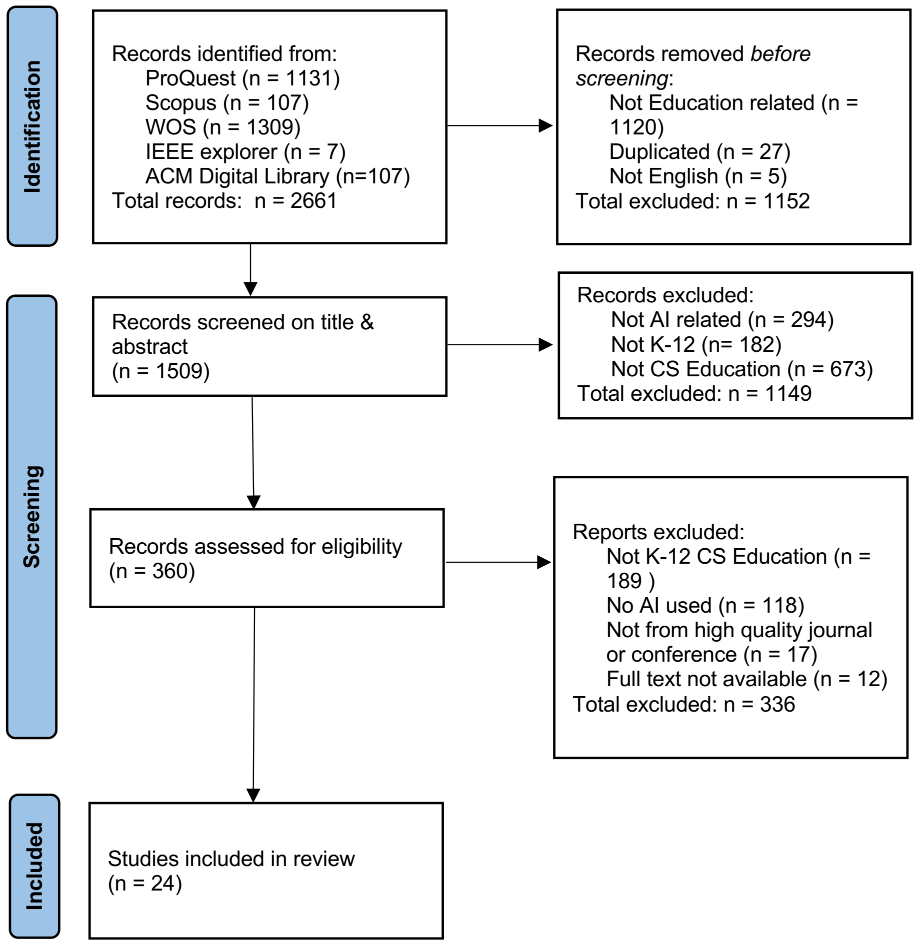


Figure 1: Prisma Diagram

Results

RQ1: What are the publication and study characteristics of K-12 CS Education with AI?

Years of Publication: For all the articles selected, 17 out of 24 was published after 2018. The notable surge in research output in recent 5 years means that studies focusing on AI experienced a substantial increase. This suggests a growing interest and recognition of the potential of AI in shaping CS education at the K-12 level.

Research Background: For the research background of these reviewed articles, shown in Figure 2. The combined visual comprises a bar chart and a pie chart detailing the distribution of articles by region and education level.The bar chart shows that the United States is the region with the most publications on the relevant topic, followed by Asia and Europe. The pie chart provides an aggregate view, showing a higher proportion of articles targeting middle and high school levels.

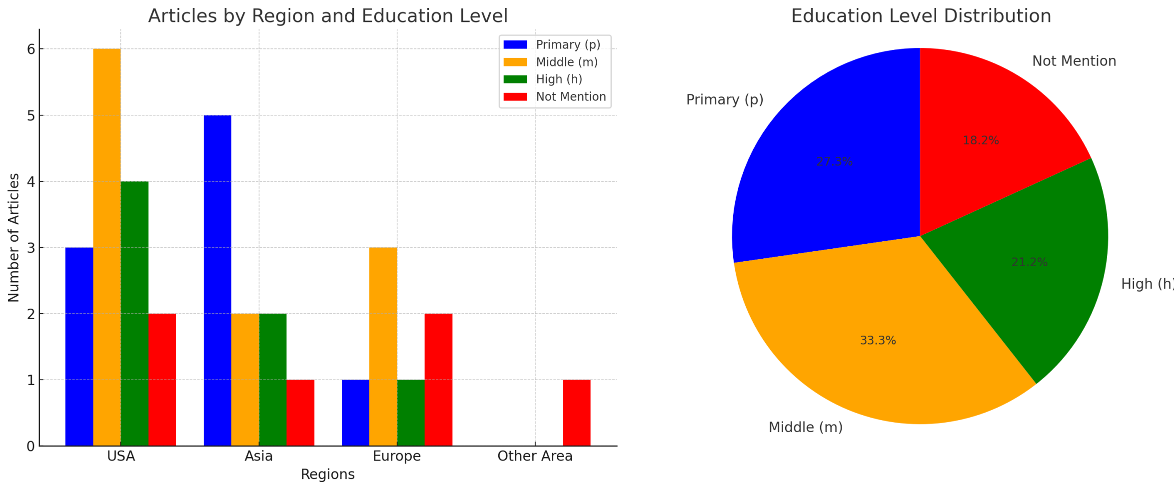


Figure 2: Areas and education levels of the selected paper

Research Types: The selected studies exhibited diverse research types. Empirical studies were prevalent, particularly experimental, and quasi-experimental designs, aimed at evaluating the impact of AI interventions on student learning outcomes and engagement levels. Additionally, case studies, interviews and theoretical studies were employed to delve into nuanced aspects of student experiences and teacher perceptions in the context of AI-enhanced CS education.

Research Focus: Many studies treat AI as learning content in CS education, while others apply AI to computer education systems. Additionally, some papers use AI methods to analyze student learning data in computer education. The research focus in K-12 CS Education included but was not limited to examining the effects of AI-supported CS teaching methods, assessing learning performance and progress using AI and process data, curriculum development, and the integration of emerging AI applications. Additionally, studies explored the the factors influencing teachers' adoption of AI, conceptions of AI of teachers and students on K-12 AI education.

RQ2: How does AI impact student learning in K-12 CS education?

To answer the question how does AI impact student learning outcomes and engagement in K-12 CS education, this study provides several perspectives on AI types, the role of AI, and student learning outcomes.

AI types

As shown in Table 3, AI types such as Machine Learning (ML), Data Mining (DM), Natural Language Processing (NLP), Computer Vision (CV), Robotics and Deep Learning (DL) are recurrent themes in K-12 CS education. Note: In Table 3, 'ITS' stands for Intelligent Tutoring Systems, 'AAS' refers to Automatic Assessment Systems, and 'PAT' denotes Programming Assistance Tools. 'VLS' represents Virtual Labs and Simulations, while 'LA' signifies Learning Analytics. The acronym 'NLP' is used for Natural Language Processing Tools, and 'CG' for Computer Programming Educational Games. Additionally, 'ML&DL Education' refers to

Table 3: AI types and AI's role

Articles	AI Types	AI's Role
[29]	ML, NLP, DM, Others	ITS, AAS, PAT, VLS, LA, NLP
[30]	CV	ITS, PAT, VLS
[12]	ML, DM	PAT, LA, CG
[31]	ML	ITS, AAS, PAT
[32]	Robotics	AI Education
[23]	ML, NLP, DM, Others	ITS, AAS, PAT, LA
[33]	CV	AI Education
[13]	Others	AI Education
[9]	ML	AAS
[34]	DL	LA
[35]	Others	PAT, CG, AI Education
[36]	DM	AAS, LA
[37]	ML, CV	AAS, VLS
[38]	ML	AAS, LA
[39]	ML, NLP, CV	AI Education
[4]	NLP, Others	ITS, AAS, PAT, LA, NLP
[40]	ML	AI Education
[41]	Others	AI Education
[42]	Others	AI Education
[43]	Others	AI Education
[44]	DM	LA
[45]	Others	AI Education
[46]	Others	AI Education
[47]	ML, DM	PAT, CG

Machine Learning and Deep Learning Education, and 'CV&SR' stands for Computer Vision and Speech Recognition Tools.

Machine Learning (ML): ML emerges as the most prevalent technology, being a fundamental approach to many AI applications as noted in[29,23].In computer education, ML is aimed at enabling programming systems to learn from student behavior or programming data. It leverages these learning outcomes to enhance the system's effectiveness, thereby better fulfilling tasks such as providing learning feedback or predicting learning outcomes. ML was used in many AI-supported programming tutoring systems to identify errors in students' solution and provide appropriate feedback to the students[29], for example,[12] designed AMOEBA to provide real time analyses of students' programming behaviors in order to support teacher in orchestrating classroom collaboration, ITAP, another programming assisting tool introduced by[31], is capable of automatically generate personalized hints for students, even when given states that have not occurred in the data before. Due to the significant impact of feedback provided by the programming environment on novice programming students, machine learning offers adaptive and timely feedback for these students[9], researchers have proposed a fuzzy-rule-based system employed to observe the students' actions and offer customized feedback using a dynamic feedback mechanism that guarantees the learner's progress through different scenarios[37]. In addition, machine learning techniques were utilized to detect behavioral events from log data and implement lag sequence analysis to extract behavioral sequences that represent the programming strategies of learners[38]. As a subset of AI, ML is also incorporated as a thematic element in CS education curricula[39,40].

Data Mining (DM): DM is utilized to discover patterns and relationships in large datasets for personalized learning and the provision of customized learning resources[29,36]. In CS education, personalized learning is gaining increasing attention. Data mining can analyze students' learning history and behaviors, providing customized learning paths and resources for each student to meet their unique learning needs.[12] designed and developed AMOEBA, which employs data mining analyses to generate real-time metrics that identify potentially successful partners and assess the effectiveness of pairings. These metrics encompass measures of participation and learning transfer, enabling the tool to support informed decision-making regarding collaborative partnerships in CS classrooms. Another automated programming assessment system APAMP apply data mining to allows students to practice repeatedly by providing immediate feedback after their programs are submitted. It also presents an analytical dashboard as a competition mechanism for students to visualize their learning performance and compare their performance with peers. A methodological framework driven by Sequential Data Analytics (SDA) has been developed and implemented to design adaptability in Digital Game-Based Learning, which aims to enhance personalized learning experiences for children in K-5 computing education[44].

Natural Language Processing (NLP): In CS education, NLP can facilitate language-based learning activities, such as programming language understanding, code summarization, and natural language-based programming, where students learn to express programming concepts in natural language[48,49].[29] outlined the use of dialogue-centric methods in AI-enhanced tutoring systems, which aid students in formulating pseudocode answers in a natural language format tailored to particular challenges. Furthermore, NLP offers an additional advantage by enabling conversational student support, leveraging knowledge representation to depict a cohort of students and their communicative dynamics during collaborative learning in CS. More recently, large language models like ChatGPT are used to assists users by clarifying intricate ideas and technologies, offering examples, and directing them to pertinent materials[50,51,52]. It also aids in identifying and solving technical issues.

Other types of AI: like CV, Robotics, DL are also found in the reviewed articles. In education, CV can be used to create interactive learning environments. Using gesture recognition, students can interact with educational content in a more engaging and intuitive way, such as manipulating 3D models of data structures or algorithms[53]. However, in CS education, CV is acting more as a learning content than a supporting technology for learning and teaching[33]. Robotics in education focuses on the design and creation of robots that can perform tasks autonomously, which has significant implications for manufacturing, healthcare, and service industries. As described in[32], virtual robotics curriculum can offer a productive learning context for K–12 CS courses that aim to teach generalizable programming knowledge and skills. While Deep Learning, a subset of machine learning, is specifically mentioned in[34] for its role in complex data analysis and feature extraction.

AI roles

In K-12 CS education, AI predominantly serves in the facilitation of AI-centric pedagogical modules (AI Education) and the provision of programming assistance tools (PAT). Concurrently, the domains of learning analytics (LA) and automatic assessment systems (AAS) feature prominently. Further incorporation of AI is observed in intelligent tutoring systems (ITS) and virtual labs and simulations (VLS), which enhance the interactive learning environment. The utilization of computer programming educational games (CG) and natural language processing tools (NLP) exemplifies the prospective roles that AI could adopt in advancing educational methodologies.

AI Education: With the development of AI in the last decade, AI modules have become an integral part of K-12 computer education.AI education improves students' technological literacy[13], enabling them to understand and evaluate AI technologies and their applications in daily life[33]. At the same time, AI education encourages logical thinking, problem-solving skills, and creative thinking[32,35], and also teaches students how to critically assess the social and ethical impacts of AI and develops a responsible attitude toward technology[39].

Programming assistance tools (PAT): In K-12 computer education, the specific application of programming assistance tools (PAT) is considered an integral part of the teaching and learning process. According to the literature[29,23], most existing learning systems aim at analyzing student's solutions and providing feedback, which serves as an important means for improving programming skills. These applications of PAT tools not only improve the effectiveness of teaching and learning[30], but also, through data-driven insights and personalized learning support, greatly promote student interest and achievement in CS[12,31].

Learning analytics (LA): LA involves measuring, collecting, analyzing, and reporting data about learners and their contexts, for purposes of understanding and optimizing learning process and the learning environments[34,23]. [38] applied learners' performance in programming tasks using data of programming behavioral events and behavioral sequences to predict programming performance in a block-based programming environment and achieved a high degree of accuracy. Analysis of students' programming behaviors can help to identify and evaluate strategies that promote learning outcomes. For example, paired programming, a collaborative teaching method, can deepen the understanding of programming concepts through mutual explanations and discussions among students[12].

Adaptive Assistance Systems (AAS): Automated assessment tools have gained popularity in CS education in the past decade[29], it can provide several benefits in CS education, including increased efficiency, scalability, and objectivity in grading[4]. Utilizing AI to automatically assess student assignments and exams, these systems provide instant feedback, thus helping instructors save time and provide accurate analysis of student learning progress[23,9,36,37].

Intelligent Tutoring Systems (ITS): ITS are systems that provide personalized instruction and feedback to learners, as referenced in[29] and[31]. [30] presented the ChiQat-Tutor intelligenttutoring system, which offers an visualized environment based on students' code for learning core CS topics. In programming education, the generation of personalized hints using state abstraction, path construction, and state reification techniques can provide customized feedback based on the individual learning needs of students. This approach works by analyzing the steps a student takes in problem-solving, guiding them towards the correct solution, and creating concrete hints that facilitate learning. According to the research by[31], such personalized feedback methods can significantly improve programming education by offering students support that is tailored to their specific learning requirements. AI techniques, which have been deployed in different tutoring approaches, serve three purposes: to support adaptive navigation, to analyze student solutions, and to enable a conversation with students[29].

Virtual Labs and Simulations (VLS): VLS have also been integrated into CS education, athough not so many works are being done in the latest reviews[29,23]. These tools are instrumental in supporting visual and experiential learning methodologies. They allow students to engage in interactive simulations, which can replicate real-world scenarios or abstract CS concepts, thereby enhancing understanding and retention of key ideas[37,30].

Computer Programming Educational Games (CG): Educational games in computer programming offer an interactive and engaging approach to learning programming concepts[12,35,47]. These games often incorporate problem-solving and critical thinking elements, making learning both enjoyable and effective. By presenting programming challenges in a game format, students are encouraged to develop their skills in a playful yet educational environment, fostering both motivation and a deeper understanding of programming[47].

Natural Language Processing Tools (NLP): NLP tools in CS education have been developed to make the learning more engaging[4]. Although there are few NLP applications in the collected literature related to K-12 CS education, what can be found is that NLP tools are particularly useful in automated tutoring systems and interactive learning platforms, where they can provide immediate feedback, clarify programming concepts, and assist in troubleshooting coding errors, thus making the learning experience more accessible and efficient[4].

Student Learning Outcomes

The dependent variables assessed in the reviewed studies primarily revolved around student cognitive, affective and behavioral levels (shown in Table 4). Cognitive levels were evaluated through learning performance such as test scores[32,36], code complexity[35], and mastery of specific CS concepts and skills such as algorithmic thinking[37] and computational thinking skills[44]. Affective levels was gauged by self-reported interest, motivation[32]and attitudes[36] in CS especially conceptions of AI and AI ethical awareness. Almost all the studies show a positive result of affective level after integrating AI in classroom. Lastly, behavioral data, including process data and behaviors in the learning platform was evaluated with the help of AI tools[44].

The studies encompassed a range of research methods. Quantitative approaches were dominant, with controlled experiments and quasi-experimental designs being prevalent[36,37]. Pre and post-surveys were commonly used to measure changes in learning outcomes[40]. Additionally, surveys[32], tests[37], and behavioral tracking tools (Log data)were employed to assess engagement levels. Qualitative methods, such as interviews[32,39,41], observation and coding of students' work[32], were utilized to gain deeper insights into student experiences and perceptions.

In K-12 CS education, research involving AI as a learning subject includes studies on students,

Table 4: Variables assessed in the reviewed studies

Dimension	Depended variables	Frequency
cognitive	learning performance	2
	computational thinking	1
	algorithmic thinking	1
	sequencing skills	1
	code complexity	1
	higher order thinking tendency	1
affective	motivation	2
	learning attitude	3
	interest,	1
	identity	1
	competency beliefs.	1
	usability,	1
	extensibility	1
	deployability	1
	intention	1
	perception of learning	1
	Conceptions of AI	1
	Ease of Learning, Ease of Use, Usefulness	1
	Satisfaction	1
	AI ethical awareness, ethical reasoning, and	2
behavioral	process data(log data)	2
	behavior	1

teachers, and their collective interactions. The curriculum is designed with activities that build upon students' existing knowledge and interests to better engage them in learning about AI[13,40,41].[39] explored teachers' perception of the open and interactive e-book and their intention to continue using the e-book to teach AI, and found positive relation between two.[33] developed curriculum which was effective in teaching AI to middle school students. The curriculum provided interdisciplinary connections, structured resources, and inclusive approaches that helped educators teach AI effectively.[35] demonstrated that the Tooee extension proposed enables block-based programming environments to support the creation of complex big data and AI programs, which were previously only possible with text-based programming. Further, comparative analyses and teacher surveys have shown that Tooee offers clear advantages over other educational tools for teaching these advanced concepts, making it a valuable addition to K-12 CS education. While AI literacy involves understanding AI's capabilities for different job roles, using AI tools to solve a wide range of problems efficiently and ethically, and applying AI in various social and cultural contexts, considering the specific norms and traditions of each setting[43],[41] found that students' conceptions of AI tended to focus on programming and robotics and they had vague and basic existing knowledge of AI.

When AI is acting as a programming assiting tool, the findings of the reviewed studies revealed a positive impact of AI on both student learning outcomes and engagement in K-12 CS education[29,23,4]. Quantitative data indicated statistically significant improvements in test scores and project completion rates among students exposed to AI-driven interventions.[9] examied an adaptive immediate feedback system significantly increased students' intentions to persist in CS, improved their engagement and learning, and was well-received by students.[32] highlights the efficacy of virtual robotics as a tool for teaching programming in middle school, emphasizing the importance of structural logic in programming for deeper learning and sustained interest in CS. Qualitative data provided valuable insights into the enhanced motivation and interest levels observed in these groups.

It is observed that AI models substantially contribute to the field of data mining and learning analytics in computer education. These models are recognized for their capacity to provide profound and insightful assistance, thereby enhancing the understanding and optimization of educational methodologies and outcomes.[44] proposed a sequential data analytics driven methodological framework to facilitate children's personalized learning experience for computing education, the study shows that SDA can inform what in-game support is necessary to foster students learning and when to deliver effective support. Some studies have found that there may be differences in how different AI models perform in different contexts. For example,[34] found LSTM network-based models are more accurate and better at early predictions than other baseline models when game interaction log feature set and the external pre-learning measure feature set was used to predict performance. It also finds that features from game interactions are better predictors than pre-learning measures, and that deep learning models are particularly effective for early predictions.[38] created a majority vote model that predicts student performance in programming by analyzing their behavior and found that including behavior data increases prediction accuracy, suggesting this method is effective for understanding and improving programming education.

RQ3:What are the main effective AI-assisted strategies and tools for teaching CS in K-12 education?

Through a review of research papers focusing on AI as a tool or model rather than learning content, this study has identified the most successful approaches and techniques for applying AI in K-12 education. These findings are summarized in Table 5. The use of AI is revolutionizing CS education at the K-12 level by offering methods and resources that cater to learning needs. These AI-powered strategies and tools play a role in fostering a foundation in CS and programming among students, preparing them for future success in our technology-driven society.

Coding and Programming Platforms: Educators and researchers have developed platforms, such as Code.org[54], Scratch[55], and Tynker[56], which utilize AI to deliver dynamic coding instruction. These platforms adapt dynamically to each student's abilities, providing real-time feedback that enhances their programming experience[12,57,37,44]. By utilizing AI-assisted coding platforms like these, students are able to embark on learning journeys that facilitate an effective grasp of programming fundamentals[35,37,44].

Automatic Grading and Feedback: AI has the ability to automate the grading of coding assignments and projects. Tools such as AutoGradr and Replika reduce the burden on teachers

Table 5: Strategies used in reviewed studies

AI tool/model/strategy	Coding Platforms	Data Analysis	AI Tutors and Chatbox	Grading and Feedback	Personalized Learning	Visualize
ChiQat-Tutor [12]	✓			✓		✓
Amoeba [12]	✓	✓		✓		✓
ITAP [31]	✓	✓		✓		
DEEP STEALTH [34]		✓				
Tooee [35]	✓		✓		✓	✓
APAMP [36]	✓	✓		✓		
AIF [9]	✓			✓		
Tangible Robots [37]		✓		✓	✓	✓
Prediction model [38]		✓				
E-book [39]	✓		✓	✓		
SHGS [47]	✓			✓		
ASDA [44]	✓	✓		✓	✓	

while providing students with feedback on their code, which enhances the learning process. By saving teachers time in grading, they can redirect their focus towards offering personalized guidance to students based on their performance .

Data Analysis: AI also plays a role in data analysis tools used in K-12 CS education. Through the integration of AI algorithms, these tools can predict students' learning outcomes and assess their proficiency in skills[57,34]. Educators gain insights into students' learning progress while enabling students to conduct self-assessments.

Gamification and Personalized Learning: AI-powered gamification techniques and personalized learning platforms are emerging as tools for K-12 CS education. These systems adapt lesson difficulty and content based on individual student progress, ensuring that each student learns at their pace.Gamification in settings has revolutionized the way CS concepts are taught. By incorporating elements of games, learning environments become engaging and enjoyable, motivating students to explore coding and programming with enthusiasm. One notable advantage is the ability to personalize learning pathways based on each student's strengths and weaknesses, providing a tailored educational experience[58,59,60].

Visualization: To make abstract CS concepts accessible, AI-powered tools have introduced visualizations. For instance, visual programming languages like Blockly leverage AI to teach coding through blocks. These visual representations enable an understanding of programming logic and algorithms, beneficial for younger learners[12,35,37]. By making coding visually intuitive, these AI-driven visualizations empower students to grasp concepts easily[61].

AI Tutors and Chatbots: The integration of AI tutors, chatbots, or voice assistants, like IBM's Watson, into K-12 CS education is becoming increasingly common. These AI-driven helpers can answer students' questions, provide explanations, and offer assistance with CS concepts[47,23]. AI tutors are especially valuable when students face programming challenges or need clarification on topics[30]. Having access to AI tutors boosts students' confidence in their abilities and encourages an independent approach to learning[31].

Discussion and Conclusion

This review carefully examined how AI has been used in K-12 CS Education between 2013 and 2023. By analyzing articles, we identified themes and explored the different ways AI can transform teaching and learning in this field. Our analysis revealed the potential of AI applications to revolutionize approaches and methods, from personalized learning experiences to automated assessments. However, our investigation also emphasized the need for research and development. It's important to consider concerns, uphold rigorous methodologies, and ensure that educators are equipped with AI skills. As AI continues to shape education, it's crucial for stakeholders to incorporate the insights gained from this review in order to improve outcomes and prepare students for a technology-driven world. This study reveals the current status of AI in CS education in K-12 settings. There is an increasing number of studies focusing on integrating AI tools into CS education, indicating that the importance of AI is now widely recognized. Across the reviewed literature, AI emerges as a versatile tool, offering adaptive learning experiences, personalized feedback, and acting as learning analytics tools. AI-assisted strategies and tools for teaching CS include coding and programming platforms, AI tutors and chatbots, automatic grading and feedback, gamification, personalized learning, and learning content visualization. However, most of these studies are still focusing on the students' learning outcomes, with only a few papers using AI tools to analyze students' process data generated on the learning platform. In the future, more studies need to be done based on the process data generated in the learning process, such as log data, behaviors, and multimodal data (e.g., facial emotions, gestures, eye-tracking data). Thus, providing students with immediate and personalized feedback using AI will be a great strategy for teachers and learners[37,4,47].

While the reviewed studies provide valuable insights, it is essential to acknowledge certain limitations. Sample sizes and study designs varied widely, potentially affecting the generalizability of findings. Additionally, some AI ethical issues are not being considered seriously, which underscores the need for more comprehensive ethical frameworks and guidelines to navigate the complex intersection of AI and K-12 education.Moving forward, there is a need for further research in specific demographics and diverse learning environments to ascertain the broader applicability of these findings. Moreover, exploring emerging AI applications, such as NLP for language-rich CS instruction, and investigating the potential long-term impacts on student trajectories are promising areas for future inquiry.

There are limitations that need to be acknowledged when considering the findings of this literature review, even though it has been conducted with care and follows the Prisma guidelines. One important limitation is the focus of the review is on English language publications, which unintentionally excludes valuable insights and research from non-English sources. Considering that AI and education are fields where this language restriction might limit the comprehensiveness of the analysis, significant contributions from scholars who publish in languages other than English might be overlooked. Additionally, the selection criteria for this review prioritize peer-reviewed sources to ensure quality and reliability. However, this approach may result in the exclusion of insights from grey literature, conference proceedings, or emerging research. AI in education is an evolving field where innovative work may often be presented outside peer-reviewed journals. Therefore, it's important to acknowledge that some pioneering or experimental AI applications in K-12 CS education might not have been included in this review due to these limitations.

References

[1] Hugo Montiel and Marcela Georgina Gómez Zermeño. Educational challenges for computational thinking in k-12 education: A systematic literature review of "scratch" as an innovative programming tool. *Comput.*, 10:69, 2021. URL `https://api.semanticscholar.org/CorpusID:236379536`.

[2] Shuchi Grover and Roy D. Pea. Computational thinking in k–12. *Educational Researcher*, 42:38 – 43, 2013. URL `https://api.semanticscholar.org/CorpusID:145509282`.

[3] Ugur Kale and Jiangmei Yuan. Still a new kid on the block? computational thinking as problem solving in code.org. *Journal of Educational Computing Research*, 59:620 – 644, 2020. URL `https://api.semanticscholar.org/CorpusID:228861571`.

[4] José Carlos Paiva, José Paulo Leal, and Álvaro Figueira. Automated assessment in computer science education: A state-of-the-art review. *ACM Transactions on Computing Education (TOCE)*, 22(3):1–40, 2022.

[5] Shuchi Grover and Roy Pea. Computational thinking: A competency whose time has come. *Computer science education: Perspectives on teaching and learning in school*, 19(1):19–38, 2018.

[6] Maria Knobelsdorf and Jan Vahrenhold. Addressing the full range of students: Challenges in k-12 computer science education. *Computer*, 46:32–37, 2013. URL `https://api.semanticscholar.org/CorpusID:6094963`.

[7] Cameron Wilson, Leigh Ann Sudol, Chris Stephenson, and Mark Stehlik. Running on empty: the failure to teach k–12 computer science in the digital age. *Running on Empty*, 2010. URL `https://api.semanticscholar.org/CorpusID:220884923`.

[8] E. B. Witherspoon, C. D. Schunn, R. M. Higashi, and R. Shoop. Attending to structural programming features predicts differences in learning and motivation. *Journal of Computer Assisted Learning*, 34(2):115–128, 2018. doi: 10.1111/jcal.12219. URL `https://doi.org/10.1111/jcal.12219`.

[9] S. Marwan, G. Gao, S. Fisk, T. W. Price, and T. Barnes. Adaptive immediate feedback can improve novice programming engagement and intention to persist in computer science. In *Proceedings of the 2020 ACM Conference on International Computing Education Research*, pages 194–203. ACM, August 2020.

[10] Ismaila Temitayo Sanusi and Sunday Adewale Olaleye. An insight into cultural competence and ethics in k-12 artificial intelligence education. *2022 IEEE Global Engineering Education Conference (EDUCON)*, pages 790–794, 2022. URL `https://api.semanticscholar.org/CorpusID:248699532`.

[11] Rajeev Alur, Richard Baraniuk, Rastislav Bodik, Ann Drobnis, Sumit Gulwani, Bjoern Hartmann, Yasmin Kafai, Jeff Karpicke, Ran Libeskind-Hadas, Debra Richardson, et al. Computer-aided personalized education. *arXiv preprint arXiv:2007.03704*, 2020.

[12] Matthew Berland, Don Davis, and Carmen Petrick Smith. Amoeba: Designing for collaboration in computer science classrooms through live learning analytics. *International Journal of Computer-Supported Collaborative Learning*, 10:425–447, 2015.

[13] Sarah Judd. Activities for building understanding: How ai4all teaches ai to diverse high school students. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, page

633–634, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367936. doi: 10.1145/3328778.3366990. URL https://doi.org/10.1145/3328778.3366990.

[14] Kevin Robinson, Keyarash Jahanian, and Justin Reich. Using online practice spaces to investigate challenges in enacting principles of equitable computer science teaching. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, page 882–887, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450351034. doi: 10.1145/3159450.3159503. URL https://doi.org/10.1145/3159450.3159503.

[15] Deborah Seehorn, Stephen Carey, Brian Fuschetto, Irene Lee, Daniel Moix, Dianne O'Grady-Cunniff, Barbara Boucher Owens, Chris Stephenson, and Anita Verno. *CSTA K–12 Computer Science Standards: Revised 2011*. ACM, 2011.

[16] Jiahong Su, Kai Guo, Xinyu Chen, and Samuel Kai Wah Chu. Teaching artificial intelligence in k–12 classrooms: a scoping review. *Interactive Learning Environments*, pages 1–20, 2023.

[17] R Tiwari. The integration of ai and machine learning in education and its potential to personalize and improve student learning experiences. *International Journal of Scientific Research in Engineering and Management*, 7 (2):1, 2023.

[18] Mostafa Zafari, Jalal Safari Bazargani, Abolghasem Sadeghi-Niaraki, and Soo-Mi Choi. Artificial intelligence applications in k-12 education: A systematic literature review. *IEEE Access*, 10:61905–61921, 2022.

[19] Zhonggen Yu. Visualizing artificial intelligence used in education over two decades. *Journal of Information Technology Research (JITR)*, 13(4):32–46, 2020.

[20] Georgios Karalekas, Stavros Vologiannidis, and John Kalomiros. Teaching machine learning in k–12 using robotics. *Education Sciences*, 13(1):67, 2023.

[21] Galina Deeva, Daria Bogdanova, Estefanía Serral, Monique Snoeck, and Jochen De Weerdt. A review of automated feedback systems for learners: Classification framework, challenges and opportunities. *Comput. Educ.*, 162:104094, 2021. URL https://api.semanticscholar.org/CorpusID:230554209.

[22] Marcelo Guerra Hahn, Silvia Margarita Baldiris Navarro, Luis De La Fuente Valentín, and Daniel Burgos. A systematic review of the effects of automatic scoring and automatic feedback in educational settings. *IEEE Access*, 9:108190–108198, 2021. doi: 10.1109/ACCESS.2021.3100890.

[23] Tyne Crow, Andrew Luxton-Reilly, and Burkhard Wuensche. Intelligent tutoring systems for programming education: A systematic review. In *Proceedings of the 20th Australasian Computing Education Conference*, ACE '18, page 53–62, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450363402. doi: 10.1145/3160489.3160492. URL https://doi.org/10.1145/3160489.3160492.

[24] Lorena Casal-Otero, Alejandro Catala, Carmen Fernández-Morante, Maria Taboada, Beatriz Cebreiro, and Senén Barro. Ai literacy in k-12: a systematic literature review. *International Journal of STEM Education*, 10 (1):29, 2023.

[25] Selin Akgun and Christine Greenhow. Artificial intelligence in education: Addressing ethical challenges in k-12 settings. *AI and Ethics*, pages 1–10, 2021.

[26] Weiqi Xu and Fan Ouyang. The application of ai technologies in stem education: a systematic review from 2011 to 2021. *International Journal of STEM Education*, 9(1):1–20, 2022.

[27] José Paiva, Álvaro Figueira, and José Leal. Bibliometric analysis of automated assessment in programming education: A deeper insight into feedback. *Electronics*, 12:2254, 05 2023. doi: 10.3390/electronics12102254.

[28] Alessandro Liberati, Douglas G Altman, Jennifer Tetzlaff, Cynthia Mulrow, Peter C Gøtzsche, John PA Ioannidis, Mike Clarke, Philip J Devereaux, Jos Kleijnen, and David Moher. The prisma statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration. *Annals of internal medicine*, 151(4):W–65, 2009.

[29] Nguyen-Thinh Le, Sven Strickroth, Sebastian Gross, and Niels Pinkwart. A review of ai-supported tutoring approaches for learning programming. In Ngoc Thanh Nguyen, Tien van Do, and Hoai An le Thi, editors, *Advanced Computational Methods for Knowledge Engineering*, pages 267–279, Heidelberg, 2013. Springer International Publishing. ISBN 978-3-319-00293-4.

[30] Omar AlZoubi, Davide Fossati, Barbara Di Eugenio, and Nick Green. Chic) at-tutor: An integrated environment for learning recursion. 2014.

[31] Kelly Rivers and Kenneth R Koedinger. Data-driven hint generation in vast solution spaces: a self-improving python programming tutor. *International Journal of Artificial Intelligence in Education*, 27:37–64, 2017.

[32] Eben B Witherspoon, Christian D Schunn, Ross M Higashi, and Robin Shoop. Attending to structural programming features predicts differences in learning and motivation. *Journal of Computer Assisted Learning*, 34(2):115–128, 2018.

[33] Alpay Sabuncuoglu. Designing one year curriculum to teach artificial intelligence for middle school. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '20, page 96–102, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368742. doi: 10.1145/3341525.3387364. URL `https://doi.org/10.1145/3341525.3387364`.

[34] Wei Min, Michael H Frankosky, Bradford W Mott, Jonathan P Rowe, Angela Smith, Eric Wiebe, and James C Lester. Deepstealth: Game-based learning stealth assessment with deep neural networks. *IEEE Transactions on Learning Technologies*, 13(2):312–325, 2019.

[35] Youngki Park and Youhyun Shin. Tooee: A novel scratch extension for k-12 big data and artificial intelligence education using text-based visual blocks. *IEEE Access*, 9:149630–149646, 2021.

[36] Li-Chen Cheng, Wei Li, and Judy CR Tseng. Effects of an automated programming assessment system on the learning performances of experienced and novice learners. *Interactive Learning Environments*, pages 1–17, 2021.

[37] Salomi Evripidou, Angelos Amanatiadis, Klitos Christodoulou, and Savvas A Chatzichristofis. Introducing algorithmic thinking and sequencing using tangible robots. *IEEE Transactions on Learning Technologies*, 14 (1):93–105, 2021.

[38] Qian Fu, Wenjing Tang, Yafeng Zheng, Haotian Ma, and Tianlong Zhong. Predicting programming performance by using process behavior in a block-based programming environment. *Interactive Learning Environments*, pages 1–15, 2022.

[39] X. Zhang, A. Tlili, K. Shubeck, et al. Teachers' adoption of an open and interactive e-book for teaching k-12 students artificial intelligence: a mixed methods inquiry. *Smart Learning Environments*, 8:34, 2021. doi: 10.1186/s40561-021-00176-5. URL `https://doi.org/10.1186/s40561-021-00176-5`.

[40] Ramon Mayor Martins, Christiane Gresse von Wangenheim, Marcelo Fernando Rauber, and Jean Carlo Hauck. Machine learning for all!—introducing machine learning in middle and high school. *International Journal of Artificial Intelligence in Education*, pages 1–39, 2023.

[41] Anne Ottenbreit-Leftwich, Krista Glazewski, Minji Jeon, Katie Jantaraweragul, Cindy E Hmelo-Silver, Adam Scribner, Seung Lee, Bradford Mott, and James Lester. Lessons learned for ai education with elementary students and teachers. *International Journal of Artificial Intelligence in Education*, 33(2):267–289, 2023.

[42] Chenghong Cen, Guang Luo, Lujia Li, Yilin Liang, Kang Li, Tan Jiang, and Qiang Xiong. User-centered software design: User interface redesign for blocklyndash;electron, artificial intelligence educational software for primary and secondary schools. *Sustainability*, 15(6), 2023. ISSN 2071-1050. doi: 10.3390/su15065232. URL `https://www.mdpi.com/2071-1050/15/6/5232`.

[43] Ning Wang and James Lester. K-12 education in the age of ai: A call to action for k-12 ai literacy. *International journal of artificial intelligence in education*, 33(2):228–232, 2023.

[44] Z. Liu and J. Moon. A framework for applying sequential data analytics to design personalized digital game-based learning for computing education. *Educational Technology & Society*, 26(2):181–197, 2023. URL `https://www.jstor.org/stable/48721004`.

[45] Helen Zhang, Irene Lee, Safinah Ali, Daniella DiPaola, Yihong Cheng, and Cynthia Breazeal. Integrating ethics and career futures with technical learning to promote ai literacy for middle school students: An exploratory study. *International Journal of Artificial Intelligence in Education*, 33(2):290–324, 2023.

[46] Xiao-Fan Lin, Zhaoyang Wang, Wei Zhou, Guoyu Luo, Gwo-Jen Hwang, Yue Zhou, Jing Wang, Qintai Hu, Wenyi Li, and Zhong-Mei Liang. Technological support to foster students' artificial intelligence ethics: An augmented reality-based contextualized dilemma discussion approach. *Computers & Education*, 201:104813, 2023.

[47] Florian Obermüller, Luisa Greifenstein, and Gordon Fraser. Effects of automated feedback in scratch programming tutorials. *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, 2023. URL `https://api.semanticscholar.org/CorpusID:259297729`.

[48] Laura Moreno, Jairo Aponte, Giriprasad Sridhara, Andrian Marcus, Lori Pollock, and K. Vijay-Shanker. Automatic generation of natural language summaries for java classes. In *2013 21st International Conference on Program Comprehension (ICPC)*, pages 23–32, 2013. doi: 10.1109/ICPC.2013.6613830.

[49] Thanveer Shaik, Xiaohui Tao, Yan Li, Christopher Dann, Jacquie McDonald, Petrea Redmond, and Linda Galligan. A review of the trends and challenges in adopting natural language processing methods for education feedback analysis. *IEEE Access*, 10:56720–56739, 2022.

[50] Som Biswas. Role of chatgpt in computer programming.: Chatgpt in computer programming. *Mesopotamian Journal of Computer Science*, 2023:8–16, 2023.

[51] Ramazan Yilmaz and Fatma Gizem Karaoglan Yilmaz. Augmented intelligence in programming learning: Examining student views on the use of chatgpt for programming learning. *Computers in Human Behavior: Artificial Humans*, 1(2):100005, 2023.

[52] Christos-Nikolaos Anagnostopoulos. Chatgpt impacts in programming education: A recent literature overview that debates chatgtp responses. *arXiv preprint arXiv:2309.12348*, 2023.

[53] Jaehong Lee, Heon Gu, Hyungchan Kim, Jungmin Kim, Hyoungrae Kim, and Hakil Kim. Interactive manipulation of 3d objects using kinect for visualization tools in education. In *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pages 1220–1222, 2013. doi: 10.1109/ICCAS.2013.6704175.

[54] Hadi Partovi and Ali Partovi. Code. org. *Recuperado de https://studio. code. org*, 2018.

[55] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. Scratch: programming for all. *Communications of the ACM*, 52(11):60–67, 2009.

[56] Deepak Kumar. Digital playgrounds for early computing education. *ACM Inroads*, 5(1):20–21, 2014.

[57] Wei Li Li-Chen Cheng and Judy C. R. Tseng. Effects of an automated programming assessment system on the learning performances of experienced and novice learners. *Interactive Learning Environments*, 0(0):1–17, 2021. doi: 10.1080/10494820.2021.2006237.

[58] Nikolaos Mallios and Michael Vassilakopoulos. Evaluating students' programming skill behaviour and personalizing their computer learning environment using "the hour of code" paradigm. *International Association for Development of the Information Society*, 2015. URL `https://api.semanticscholar.org/CorpusID:13858677`.

[59] Wint Yee Hnin. Personalized learning pathways using code puzzles for novice programmers. *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 327–328, 2017. URL `https://api.semanticscholar.org/CorpusID:27138843`.

[60] Brianna Dym, Cole Rockwood, and Casey Fiesler. Gaming together, coding together: Collaborative pathways to computational learning. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 2023. URL https://api.semanticscholar.org/CorpusID:257311660.

[61] P. Gough, O. Bown, C. R. Campbell, P. Poronnik, and P. M. Ross. Student responses to creative coding in biomedical science education. *Biochemistry and molecular biology education : a bimonthly publication of the International Union of Biochemistry and Molecular Biology*, 51(1):44–56, 2023. doi: 10.1002/bmb.21692. URL https://doi.org/10.1002/bmb.21692.