Computers and Education Open 9 (2025) 100266

Contents lists available at ScienceDirect

Computers and Education Open

journal homepage: www.sciencedirect.com/journal/computers-and-education-open

Computing education using generative artificial intelligence tools: A systematic literature review

Friday Joseph Agbo [a], Chris Olivia [a], Godsalvation Oguibe [b], Ismaila Temitayo Sanusi [c],*, Godwin Sani [c,d]

[a] Willamette University, United States
[b] University of Texas at San Antonio, United States
[c] School of Computing, University of Eastern Finland, Finland
[d] Federal University Lokoja, Nigeria

ARTICLE INFO

Keywords:
Computing education
Programming
Generative artificial intelligence
Large language model

ABSTRACT

Recent advances in generative artificial intelligence (GenAI) are revolutionizing computing education, causing paradigm shifts from the traditional teaching and learning technique. Studies are exploring GenAI tools in computing classes from intro to advanced topics with the aim to showcase how to reshape computing education in this new era of GenAI. This study examined the computing education research landscape to unravel how GenAI tools have been used in that domain, what are the characteristics of those studies in terms of computing topics, context, and tools, offering insights into the pros and cons for integrating GenAI in computing education based on the performance indicators reported in the literature. This study employed a systematic literature review approach to identify and analyze 78 relevant articles. The findings of this study show that educators are exploring GenAI tools in computer sciences classes from K-12 through graduate levels. Beyond programming education, GenAI has also been explored in college upper-level computer science courses such as Computer Graphics and Human-Computer Interaction. The performance analysis of these tools are presented in this study, indicating a progressive advancement from when the technology was introduced. This study also discusses learning outcomes, good practices, and potential risks to avoid when exploring GenAI, as reported in the studies, which could guide how computing educators design their instructional strategies using GenAI. This study contributes to broadening the understanding of exploring, adapting, or using GenAI in computer science education and may spark interest among educators who are unwilling to explore GenAI or may not understand how or what strategies to adopt.

1. Introduction

The use of generative artificial intelligence (GenAI) tools in education has risen across different disciplines due to its potential to transform learning in different contexts [1]. In computing education, there has been an upsurge in exploration of Large Language Models (LLM) such as ChatGPT, CoPilot, and Codex in developing curriculum for teaching programming at higher education levels. Although the question of whether to accept the use of GenAI tools in computing education and particularly in introductory programming class is an ongoing discussion. A ground-breaking work by [2] provides insightful overview from several perspectives on how GenAI is penetrating the computing education domain. Another recent study by [3] elucidates that GenAI is transforming the practices of teaching and learning of programming by using LLM-based code generation.

Notwithstanding, many educators are still contemplating whether a paradigm shift from traditional programming education to a new approach engendered by this transformation brought about by the GenAI is worth pursuing. Some are unwilling to take the risks of introducing GenAI-based tools into their computing classes since they are unsure of the outcomes. Majority of these fears are connected to lack of understanding on how to assess students' learning outcomes, especially if they were allowed to use GenAI tools. This includes the impact of the GenAI tool on academic integrity and student cheating behaviors [4].

While there are mixed feelings about the integration of GenAI tools in computing education, many educators were bold to take the risk of introducing varied GenAI tools in their classes and reported them in published articles. Majority of these studies are exploratory research

* Corresponding author.
E-mail addresses: fjagbo@willamette.edu (F.J. Agbo), colivia@willamette.edu (C. Olivia), godsalvation.oguibe@my.utsa.edu (G. Oguibe), ismaila.sanusi@uef.fi (I.T. Sanusi), godwin.sani@fulokoja.edu.ng (G. Sani).

https://doi.org/10.1016/j.caeo.2025.100266
Received 7 October 2024; Received in revised form 3 June 2025; Accepted 11 June 2025
Available online 21 June 2025

being conducted to demonstrate how to integrate LLM and GenAI-based tools in teaching computing topics in classrooms. This exploration continues to rise as the GenAI technology keeps advancing with many enticing promises to revolutionize many sectors. What is more? There is a need to continuously examine the literature through the lenses of the literature review to broaden the understanding of educators about what has been reported that showcased the integration of GenAI tools in computing education. This understanding can also provide a roadmap for educators to delve into the use of GenAI in the classroom.

While existing research on GenAI in computing education has predominantly focused on introductory programming courses, this narrow scope risks overlooking critical pedagogical and practical challenges unique to broad computing disciplines. Studies reveal that introductory courses often emphasize basic syntax and problem-solving frameworks, where GenAI tools such as ChatGPT and Copilot excel at generating code snippets or debugging simple errors [2,3]. However, broad computing education, such as distributed systems, software engineering, web development, or human–computer interaction, requires students to synthesize complex architectures, optimize performance, and navigate ethical dilemmas in real-world applications, tasks that demand higher-order cognitive skills. A systematic literature review focusing on broad computing contexts could uncover patterns in how GenAI either enhances or hinders learning outcomes in these settings, identify gaps in tool reliability for domain-specific tasks, and inform frameworks for teaching AI literacy tailored to advanced competencies. In addition, understanding how GenAI-based tools have been explored in computing education will help us gain insight into what have been reported as potential risks to avoid, surprises to expect, and learning gains that could motivate both learners and educators. To address this gap, this study aims to provide insights into how GenAI tools have been integrated in computing education classes broadly, setting out the following research questions to be addressed.

RQ1: How have GenAI tools been used for teaching and learning in computer science education?

In dealing with this research question, the study will try to provide answers to the following sub-questions:

- Which educational level context are the studies on GenAI in computing education targeted at?
- What topics in computing education are being taught using GenAI?
- Where are the authors using GenAI in computing education?
- Which GenAI tools are being explored to foster computing education?
- What methodological approaches are used to introduce GenAI into computing education?
- What are the characteristics of the sample size reported in articles that utilize GenAI in computing education?

RQ2: What are the pros and cons of integrating GenAI tools in computing education as reported in the articles?

2. Background and related work

Generative Artificial Intelligence (GenAI) traces its origins to the 1960s with the development of ELIZA, the primitive pioneering example of a conversational agent [5]. The design of ELIZA was influenced by earlier work [6], which laid foundational principles for natural language understanding within specific domains. A generative model is a mathematical framework that describes the underlying probability distribution of observable and non-observable events [7]. Unlike discriminative models, which focus on recognizing patterns and making predictions based on existing data, generative models aim to create new, believable examples from the learned experience [8]. The wide adoption of GenAI can be attributed to the emergence of ChatGPT, which was released in November 2022 [9] with some of its common attributes including language understanding, content generation, including customer support, medical diagnosis, legal analysis, and financial forecasting.

GenAI has evolved and keeps evolving. Various GenAI tools, such as Google's Bard, Microsoft's Bing Chat, OpenAI's GPT-3, Amazon's Lex and Meta' LLaMA rely on different aspects of artificial intelligence and natural language processing. For example, Bard and Bing Chat use transformer-based architectures along with large language models to generate responses that resemble human-like ones [10]. In contrast, GPT-3 leverages the idea of "self-attention" while scaling up its linguistic models to increase understanding capabilities and generation skills. These trends in GenAI evolution pave the way for their adoption in more facets of human endeavor.

2.1. Generative AI in education

The impact of GenAI on education is significant as it can augment both the teaching and the learning experience. Encouraging findings have emerged from various research studies that investigated its educational possibilities over time; for example, chatbots powered by GenAI technology offer personalized feedback and guidance to students in their study journey [11]. Furthermore, GenAI has shown an immense potential to generate interactive lesson plans or assignment evaluations [12], reducing the workload of teachers while enhancing the quality of the instructional standards simultaneously.

However, the role of GenAI, for example, ChatGPT, in education remains a debatable topic as educators and researchers grapple with the varied ways it can be integrated into the learning process without circumventing the learning outcomes. It has been argued that ChatGPT has the potential to reshape pedagogical approaches by offering on-demand assistance, facilitating adaptive learning experiences, and creating interactive and engaging learning experiences [13]. According to Prather et al. [2], the Large Language model provides several opportunities and challenges in education which include; new instructional approaches, academic integrity concerns, and replications and redundancies in contents.

GenAI has been leveraged to foster project-based learning and enable innovative team collaboration among students [14]. Additionally, research shows that GenAI can aid the education of special needs learners [15]. However, concerns over its impact on educators, tasks as well as student academic achievement require attention [16]. Moreover, it is essential to uphold ethical standards for implementing GenAI in schools concerning issues around equity and confidentiality. In fostering an inclusive learning environment that addresses bias in assessment of learning outcomes in the era of GenAI, a recent study presents several approaches including self-regulated assessment strategy [17,18]. According to Xia and colleagues [17], self-assessment is a strong strategy that could support students' learning outcomes, thereby allowing them to learn from the mistakes and errors of a learning task that was suggested by GenAI. In addition, another study by Shen et al.'s [19] reported that prompt engineering significantly improved ChatGPT's performance when evaluated on data science assignments.

2.2. Programming education in the era of generative AI

One of the early studies on GenAI in programming education was conducted by Muller and colleagues [20] who delved into the intersection of human-centric themes with generative artificial intelligence, emphasizing the importance of comprehending collaboration patterns between AI systems and humans. They assert that there is a unique opportunity for GenAI systems to augment human efforts in teaching programming. Similarly, Prather et al. [2] demonstrated that recent advances in GenAI technology are revolutionizing computing and society at large. The researchers recommended utilizing LLM, which facilitates natural language production and comprehension along with source code synthesis, paving the way for investigating possible applications of GenAI within computer classrooms [2].

In addition, Cambaz and Zhang [3] conducted a study to validate the notion that recent AI-driven code generation models can potentially transform programming education. Their systematic review of 21 papers found that AI code generators can be an assistive tool for learners and instructors if the risks are mitigated. Similarly, according to Kazemitabaar et al. [21], LLMs enable natural language programming that can perform code-to-code operations such as code completion, translation, and repair, in addition to language-to-code functions like code explanation. By harnessing the potential of GenAI in programming education, the way coding is taught and learned can be revolutionized. With AI-driven code generation, natural language programming, and collaborative learning, a more inclusive, and effective programming education can be created to prepare the next generation of coders for success.

Meanwhile, GenAI in computing education aligns closely with constructivist and cognitivist learning theories, offering mechanisms to scaffold knowledge construction and optimize cognitive processing. Constructivist principles are evident in the ability of GenAI to facilitate active and personalized learning through code generation, interactive problem solving (for example, debugging simulations), and collaborative projects, enabling learners to build knowledge through hands-on experimentation and reflection [22–24]. For instance, tools such as ChatGPT foster critical thinking by prompting students to refine code iteratively [2,25–27]. Cognitively, GenAI can reduce extraneous cognitive load by automating routine tasks (for example, syntax correction), allowing learners to focus on higher-order concepts such as algorithm design [28].

While existing research illuminates GenAI's transformative potential in programming education, computing education spans diverse domains — from cybersecurity to human–computer interaction — each with distinct pedagogical needs and technical challenges that introductory programming frameworks may not generalize. A systematic review with broader scope is critical to uncover how GenAI's role shifts across subfields.

This paper systematically reviews the evolving landscape, focusing on how GenAI intersects with CS topic domains, influences teaching and learning strategies, shapes learning outcomes (highlighting both advantages and drawbacks), and prompts critical reflections and future directions for the field. The conceptual diagram presented in Fig. 1 visually synthesizes these multifaceted interactions. By mapping these relationships, the conceptual diagram underscores the rationale for this study, which is to provide a comprehensive, evidence-based understanding of GenAI's role in computing education, ensuring that its adoption is guided by pedagogical best practices and a commitment to fostering inclusive, reflective, and forward-looking teaching and learning experiences.

3. Methodology

This study maps the existing knowledge from studies that explored GenAI tools in teaching and learning computing education in a formal educational context. The study's goal is to uncover relevant knowledge from published articles regarding the experiences of educators who have used these new technologies in computing classes. A systematic literature review methodology guided by the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) framework [29] was used in this study. As depicted in Figure 1, this study collected data from relevant sources and followed PRISMA guidelines to identify the data, screen for relevance, and analyzed those articles that met the inclusion criteria.

3.1. Article identification process

The search strategy demonstrated in this study is systematic rather than exhaustive. First, the search for the article was targeted at four widely used databases for publishing and indexing computing education studies. The databases include ACM Digital Library, Scopus, Web of Science, and IEEE Xplore. Next, we identified a list of keywords that could be used to query the databases for data collection. Each database provides means for customizing the search strings which differ from one another in terms of design but are somewhat similar in principle. The strategy used in search string combination was informed by past literature [2,30], carefully constructed the search strings using logical operators. For example in Scopus, the search strings combination with logical operators is (*"Large Language Model" OR "LLM" OR "GenAI" OR Generative Artificial Intelligence"*) AND ("Computing Education" OR "Computer Science Education" OR "Programming Education"). Although the authors are aware of possibilities of articles that are archived in open-source forums such as arXiv, which do not undergo peer reviewing process, however, we were deliberately interested in collecting articles that underwent rigorous process of peer-reviewing and published in reputable outlets. The search condition did not include any restriction such as date range but rather allowed the algorithm to search the full text. The search was conducted on May 13 2024 and the results returned from each database considered in this study are shown in Fig. 2.

3.2. Article screening

The total data collected from the databases were over 374 articles from which there were some duplicate data which were removed. Then, the authors designed a Google sheet to store articles that met the initial inclusion and exclusion criteria after skimming the title, abstract, and keywords of each article. The initial inclusion criteria were that the paper must be written in English language and published in either journal or conference proceedings. Table 1 presents the list of inclusion and exclusion criteria developed by the authors based on the focus of this study to guide the choice of relevant articles to be analyzed. During the first screening stage, three authors were assigned to populate only articles that met the initial criteria on the Google sheet created and shared among the authors. This initial data screening produced 106 data. At this point, the authors developed a list of outcome domains - following PRISMA guidance [29] - to guide data synthesis to address the research questions. For example, the list of domain outcomes based on this study objective includes *article title, study aim, targeted audience, computing topic, educational context, GenAI tool, reported outcomes, publication venue, study methodology, and authors affiliation*. This list formed the columns of the Google sheet created for data synthesis. Three authors divided the data among themselves to read the articles thoroughly and code them into the new Google sheet according to the domains (see publishedGooglesheet).

Each assigned author independently read the paper in more detail to decide whether the article discussed the use of or explored GenAI tools in computing education. If an article is deemed not relevant, the author would highlight it red so that the team can meet afterward to agree on the decision.

3.3. Data validity and eligibility

After the initial screening was completed, the three authors met again for intercoder reliability and validity. The meeting was held on Zoom and authors went through the data to specifically screen the data highlighted irrelevant and agreed on their eligibility. During this process, a few of the articles were reconsidered while some were removed upon agreement between the authors. This process also eliminates bias that may occur in the selection of articles that are included

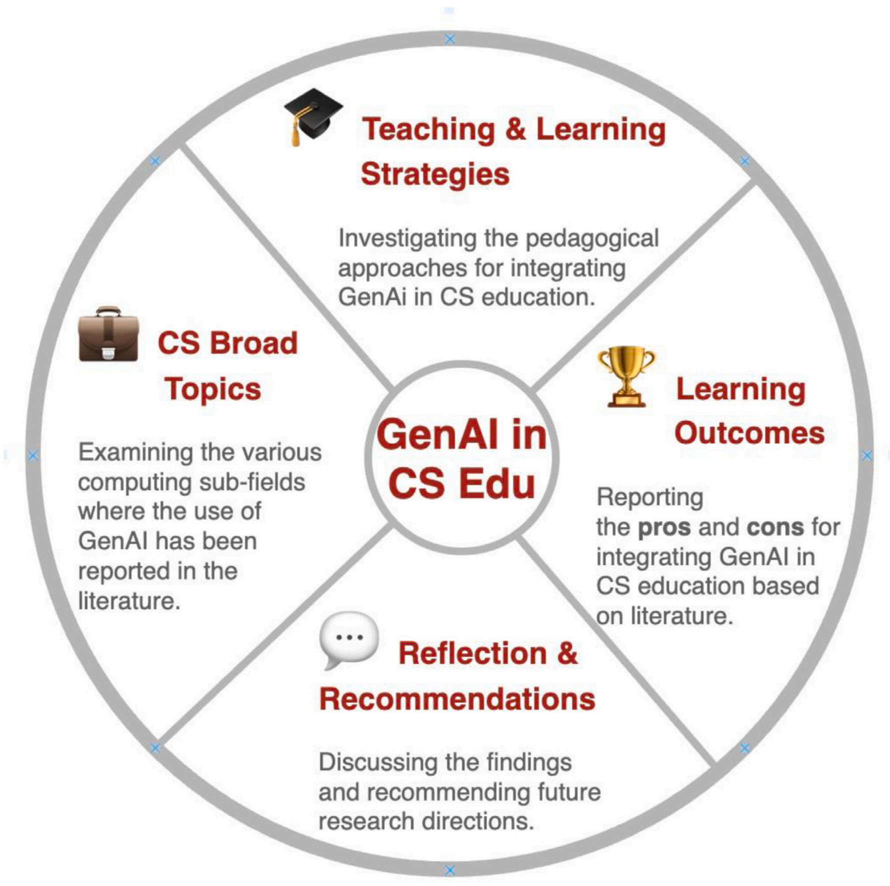

Fig. 1. Conceptual diagram illustrating the interplay between GenAI and various facets of computing education reported in this study.

Table 1
Inclusion and exclusion criteria.

Inclusion criteria	Exclusion criteria
Papers written in English language	Papers that are not written in English language but other languages such as Spanish, Russian, etc.
Research articles published in journal or conference proceedings and are peer-reviewed	Paper classified as "proceeding", "keynote", "panel", "News", which did not undergo peer-reviewing process.
Papers that discusses LLM or AI models, tools, approaches, and designs to foster computer science education in formal setting	Papers that reported mere opinions and concerns about use of GenAI tools in programming and computer science education or failed to give expert perspective based on experience or experiment with or explored it in any form.

in the analysis. Consequently, 75 papers emerged as the total number of articles that satisfies the eligibility criteria.

Furthermore, we employed the concept of snowballing to find data that we may have missed out during the data selection stage. This process was mainly conducted by looking through the "related work" section and list of references of randomly selected articles that we deemed relevant to our study [2]. The snowballing process identified a number of articles, however, only three were relevant to our study and were included which resulted in 78 articles analyzed in this study.

4. Results

This section presents the study findings by first elucidating the characteristics of the data being synthesized before delving into addressing the research questions formulated for this study. The analysis of important characteristics of the data will provide context to the overall findings. Thus, we present the annual article production, venues where the articles were published, and the corresponding author's affiliation country.

The analysis depicted in Fig. 3 shows that there were four articles on GenAI for computing education published in 2022 [31–34]. Although OpenAI was launched in 2022, there may be more articles published in that year in venues such as arxiv.org which our study did not cover due to the lack of rigorous peer-review. These four articles could be considered to be among the first research that explored the use of GenAI in computing education. The corresponding authors in these early articles are from the United States, New Zealand, and Finland. Noticeably, two of the articles were presented at the 2022 International Computing Education Research (ICER) and published in the same year's proceedings for the ACM conference on ICER. All the articles have been cited over 150 times as of the analysis of this study.

Furthermore, there was a rapid increase in publication regarding GenAI in computing education between 2023 and first quarter of the

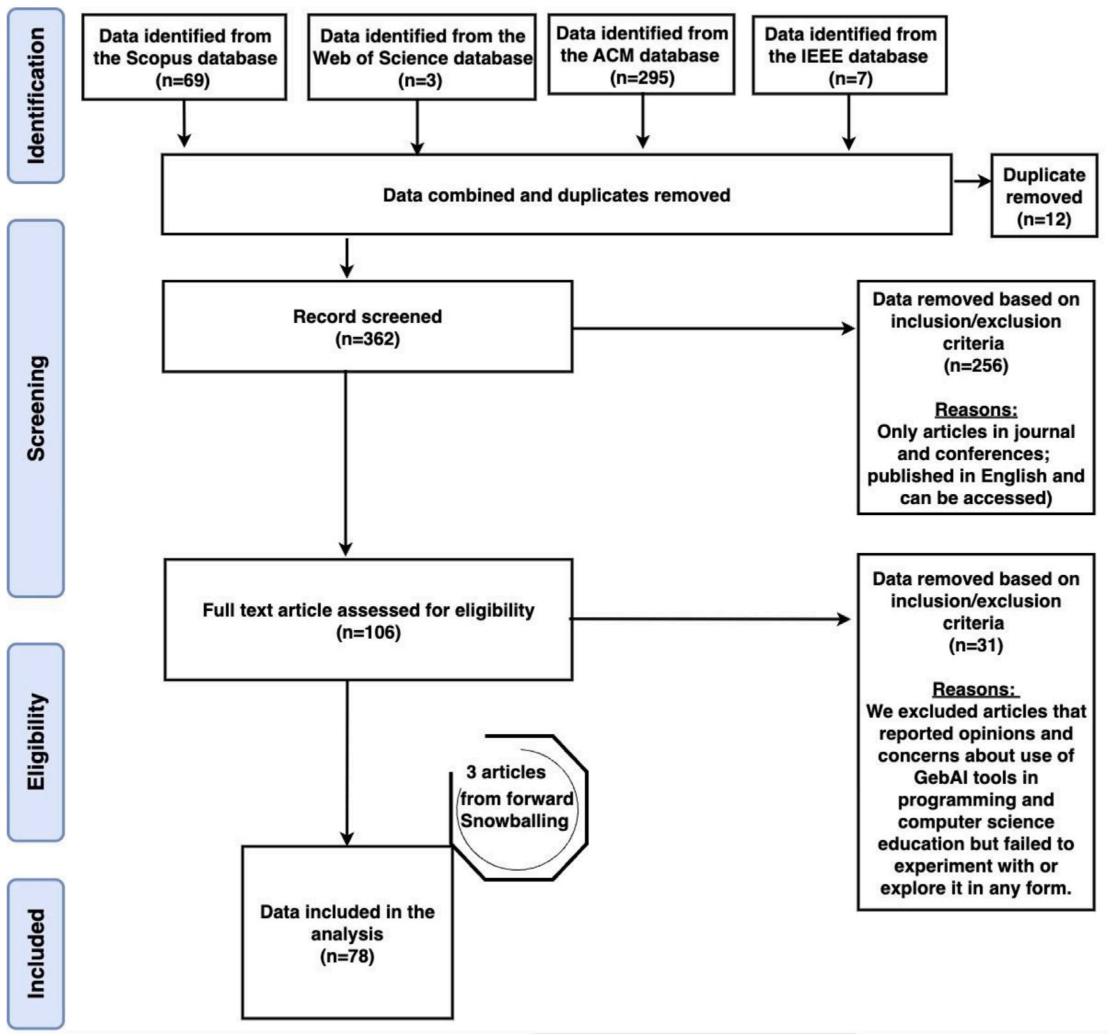

Fig. 2. Data collection procedure following PRISMA guidelines.

year 2024. We found 36 articles published on this topic in 2023 and 39 articles published in the first quarter of 2024 alone, respectively. This suggests that the growth rate of publication in 2024 regarding the exploration of GenAI tools in computing education is projected to be about five times higher than the previous year.

Regarding the venues where the articles reviewed in this study are published, the analysis in Table 2 shows that only 7 (i.e. 9%) of the articles were published in journal outlets while 72 (i.e., 91%) of the articles were published in conference proceedings. This typically shows that many of the articles have been discussed at scholarly forums which could spark educators' curiosity and interest to delve into exploring the use of the GenAI in computing education.

RQ1: How have GenAI tools been used for teaching and learning in computer science education?

As explained above, we attempted to provide answers to the first research question by addressing a list of sub-questions.

Which educational level context are the studies on GenAI in computing education targeted at?

First, we analyzed the educational context studies on GenAI in computing education were targeted at and found that most studies (91%) are exploring GenAI tools for computing education at the undergraduate level (i.e., colleges and universities) compared to K-12 (6%) or other graduate (3%) settings. However, we found a paper that explored both graduate and undergraduate settings (Vaithilingam et al. 2022).

What topics in computing education are being taught using GenAI?

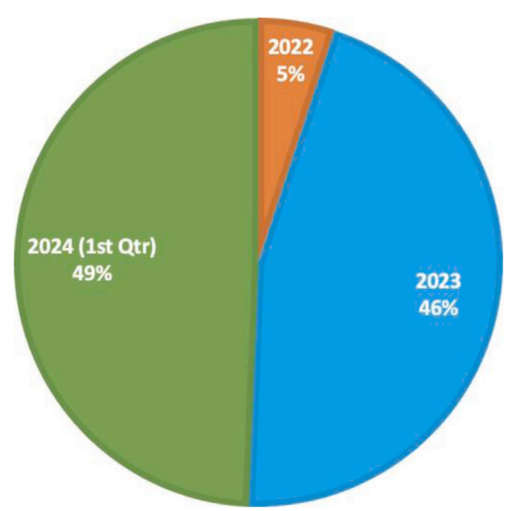

Fig. 3. Articles introducing GenAI tools in computing education published between 2022 and first quarter of 2024.

Table 2
Popular Venues publishing research on GenAI in computing education.

Venues	Frequency
Applied Sciences (Switzerland)	1
IEEE Access	1
IEEE Transactions on Learning Technologies	1
International Journal of Engineering Pedagogy	1
Journal of Computing Sciences in Colleges	1
Journal of Information Processing	1
Mathematics	1
International Conference on Intelligent User Interfaces	1
IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC	1
Asia Service Sciences and Software Engineering Conference	1
International Conference on e-Society, e-Learning and e-Technologies	1
United Kingdom and Ireland Computing Education Research	1
Annual Conference on Information Technology Education	1
ACM International Conference on Design of Communication	1
International Conference on Software Engineering: Software Engineering Education and Training	1
European Conference on Software Engineering Education	1
Conference on Computing Education Practice	1
International Conference on Intelligent User Interfaces	1
International Conference on Computer Supported Education, CSEDU	2
Learning Analytics and Knowledge Conference	2
Annual ACM India Compute Conference	2
The AAAI Conference on Artificial Intelligence	2
ACM Conference on Global Computing Education	2
Brazilian Symposium on Human Factors in Computing Systems	2
Frontiers in Education Conference, FIE	3
Koli Calling International Conference on Computing Education Research	3
Conference on Innovation and Technology in Computer Science Education, ITiCSE	4
ACM Conference on International Computing Education Research, ICER	5
Conference on Human Factors in Computing Systems, CHI	6
Australasian Computing Education Conference	8
ACM Technical Symposium on Computer Science Education, SIGCSE	19

Table 3
Country affiliation of corresponding authors publishing articles on GenAI in computing education.

Country of corresponding author	Number of articles
Australia	2
Austria	1
Bangladesh	1
Brazil	2
Canada	2
Czech Republic	1
Finland	3
Germany	2
India	5
Iran	1
Ireland	2
Japan	3
New Zealand	6
Portugal	1
Republic of Korea	1
Saudi Arabia	1
Spain	1
Sweden	2
The Netherlands	1
United Kingdom	1
United States	38

Table 4
Topics in computing education where GenAI is being explored.

Computing education topics	Frequency of occurrence
Computer Graphics	1
Computing education	33
Data Structures and Algorithms	1
Data science education	2
Human–computer Interaction	2
Object-oriented programming	1
Programming education (Introductory programming)	34
Software development	1
Software Engineering	1
User Experience design and Web Development	1
Web software development	1

Regarding the topics in computing education scholars are exploring using GenAI, this study reveals that educators first began investigating how LLM models such as ChatGPT could generate codes to address programming problems and examined its performance against students' solution. Hence the domain of programming education witnessed the early attempts by educators to integrate GenAI. As shown in Table 4, a significant number of articles (n=28) clearly indicated the use of GenAI tools in teaching and learning of programming, particularly in the introductory programming class. Furthermore, this study found that a significant number of educators are also exploring GenAI tools to solve computer science questions in general. Although these questions could be programming related, they were also broadly applied. For example, Joshi et al. [35] conducted a study to examine the strengths and weaknesses associated with the utilization of ChatGPT as an educational tool to solve examination questions from core undergraduate computer science subjects including Data Structures and Algorithms (DSA), Operating Systems (OS), and Database Management Systems (DBMS). It is interesting to note that a few articles also explored the use of GenAI in topics such as human–computer interaction [23,36] and data science education [14,19].

This finding suggests that there are emerging studies that showcases the integration of GenAI into higher level computing education classes such as software engineering and development [37], Users Experience (UX) design [38], and computer graphics [39,40].

This study also found that educators (teachers) are mostly exploring the use of GenAI in computing education (as shown in Fig. 4) for several reasons. For example, some teachers are trying out their prior examination questions on programming or other computing topics with GenAI tools to evaluate how they perform compared to students [39,41]. In addition, some educators are integrating LLM models into an existing system such as learning management systems to facilitate teaching and learning of computing topics. For example, Jin et al. [42] integrated an LLM model into a chatbot to teach algorithm design. On the other

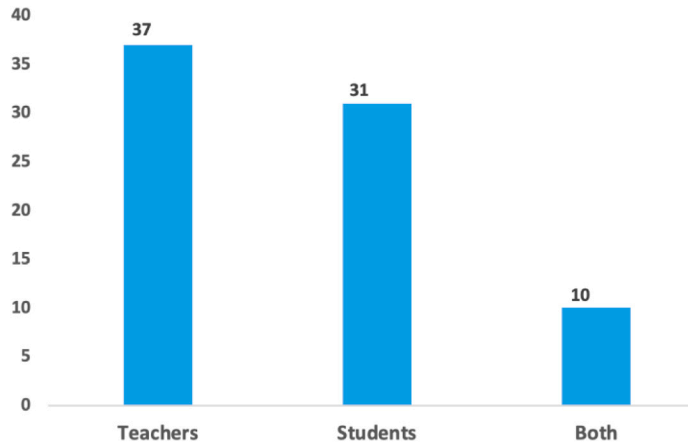

Fig. 4. Main categories of users exploring GenAI in computing education.

hand, some educators investigate the performance of GenAI by allowing students to use them directly. For example, Bhalerao [43] reported on the experience of guiding the students to use ChatGPT and GitHub copilot to learn programming. The author found that the students reported higher understanding of concepts and managed to recover from error with the help of these GenAI.

Where are the authors using GenAI in computing education?

The study analysis also shows that 49.4% of the articles reviewed in this study have their corresponding author's affiliation from the United States. Other popular countries where authors are exploring GenAI in computing education include New Zealand (7.6%), India (6.3%), and Finland and Japan (3.8%), respectively. Similarly, there are emerging studies from Australia, Brazil, Canada, Germany, Ireland, and Sweden (2.5%), respectively. Table 3 shows a few other countries such as the United Kingdom, Saudi Arabia, and more with at least one article emanating from those countries.

Teachers mostly use GenAI to facilitate the development of teaching resources, design automation of assessment processes, and to develop learning support for learners in the form of teaching assistants. On the other hand, the students are mostly guided by teachers to engage with the tools for learning purposes rather than self-regulated learning.

Which GenAI tools are being explored to foster computing education?

This study investigated the articles to uncover which GenAI tools are used to foster computing education and found dozens of tools reported in the literature. We also delved into analyzing the performance of these tools in the light of reoccurring terms found in the selected articles [3]. Table 5 presents the list of GenAI tools, the number of articles that reported to have explored them, and the performances. As the results reveal, ChatGPT (n=50) is the most popular GenAI tool used in computing education. ChatGPT-3.5 and ChatGPT-4.0 are the most reported versions of the tools. These two versions of GenAI have been generally reported to perform well on code generation, however, ChatGPT-4.0 performs better than ChatGPT-3.5. Obviously, this is due to the advancement OpenAI (developers of ChatGPT) have made to the model between 2022 and early 2024. The advancement has also caused the LLM tool to gain improved accuracy and better personalization of responses being generated by the tool.

Besides ChatGPT, other popular LLM tools used to support teaching and learning of computing education include Codex, GitHub Copilot, and OpenAI's text-davinci-003. Furthermore, there were a couple of studies that utilized a combination of these tools. For example, ChatGPT plus GitHub copilot [43], ChatGPT + Codex [44], and CS50 Duck and ChatGPT-4 [45]. Interestingly, there are studies that integrate the popular LLM model with other independent solutions. For example, Kuramitsu et al. [46] integrated ChatGPt into Jupyter notebook application to develop a GenAI tool called KOGI that allows students to prompt the tool with codes with errors in order to receive help on how to fix them.

What methodological approaches are used in exploring GenAI for computing education?

This study found that most studies (n=23) reported to have used experimental design as the methodological approach for exploring GenAI in computing education. This finding is not surprising owing to the fact that the use of GenAI is a new phenomenon since OpenAI recently launched their first LLM in 2022. Other popular methodologies as shown in Fig. 5 include mixed method and exploratory design.

It makes sense to apply a mixed method or exploratory design to examine a new paradigm engendered by artificial intelligence in educational context to unravel hidden knowledge regarding how it should be used, and unveil what works and what does not work.

What are the characteristics of the sample size reported in articles that demonstrate GenAI in computing education?

Most of the articles reviewed in this study are empirical research that reported findings in some forms. Thus, we analyzed the sample sizes reported in the articles to gain a deeper understanding of the demographics of the population.

It is worthy to note that some of the studies utilized samples other than human populations in conducting their experiment. While some studies used sample questions and problem sets from a programming class to conduct their experiments with GenAI tools, others used both human and generated texts, thus having huge sample sizes as seen in Fig. 6. For example, MacNeil et al. [28] conducted a study with 964 students to collect 2980 data responses used to examine the performance of GPT-3 and GPT-4 regarding their capabilities in detecting logic error in C programming language and providing friendly explanations to novices.

RQ2: What are the pros and cons of integrating GenAI tools in computing education as reported in the articles?

The second research question tries to examine the potential benefits and challenges of using GenAI in computing education as reported in the articles. Table 6 presents a list of potential gains reported by the researchers for using GenAI in computing education. Most of the studies reported that the use of GenAI tools helps the students to achieve a better learning experience in terms of personalizing the support they receive. For example, the students could better understand programming error messages as they prompt an LLM tool based on its previous response to a task. In addition, the students could obtain immediate feedback from a GenAI tool as compared to the time it takes them to access help from the instructor or course assistant. A number of studies reported that GenAI tools are capable of identifying errors in codes and provide useful suggestions on how to fix them.

On the teachers' part, the use of GenAI tools has helped them to prepare their teaching and learning resources for computing education,

Table 5
GenAI tools and performance based on indicators reported in the articles.

GenAI tool (Occurrence)	Code generation	Personalized Feedback	Improved accuracy	Code interpretation	Error identification	Remark
ChatGPT-3/ChatGPT-4 (50)	xx	xx	xx	x	x	Good at program tracing; limited in manipulating images, shapes and files; verbose; hallucination
Codex (7)	xx		x	–	–	Good at solving Parson problems; require prompt engineering skill; limited in manipulating complex domain problems
GitHub Copilot (3)	xx	x	xx			Code generated may be formatted poorly
OpenAI's text-davinci-003 (3)	x		–			Good at program tracing; limited in manipulating images, shapes and files; verbose; hallucination
Piazza forum and GPT-4 (1)	x	x				Limited in manipulating complex problems
SQL Query LLM (1)	x	x				
WorkedGen (1)	x	x		x		
KOGI and GPT-4 (1)	x					
KOGI, ChatGPT + Jupyter (1)	x	x		x	x	
OpenAI's GPT-3.5 Turbo (1)	x	x	–	–	x	Formative rather than summative feedback
CS50 Duck and GPT-4 (1)	x	x				
Code-davinci-002 (1)	x	x				Require prompting engineering skill
ChatGPT and Codex (2)	x	x	x			Poor code formatting
ChatGPT and Github Copilot (2)	x	x		x	x	Formative rather than summative feedback
AlgoBo LLM chatbot (1)	x		x			
AskGPT (1)	x					Limited in manipulating complex problems, require prompt engineering skill
Bard and GPT-4 (1)	x	x				Hallucination/Fabulation and Occasional refusal

Note the performance keys: Above average (xx); Average (x); Below average (-)

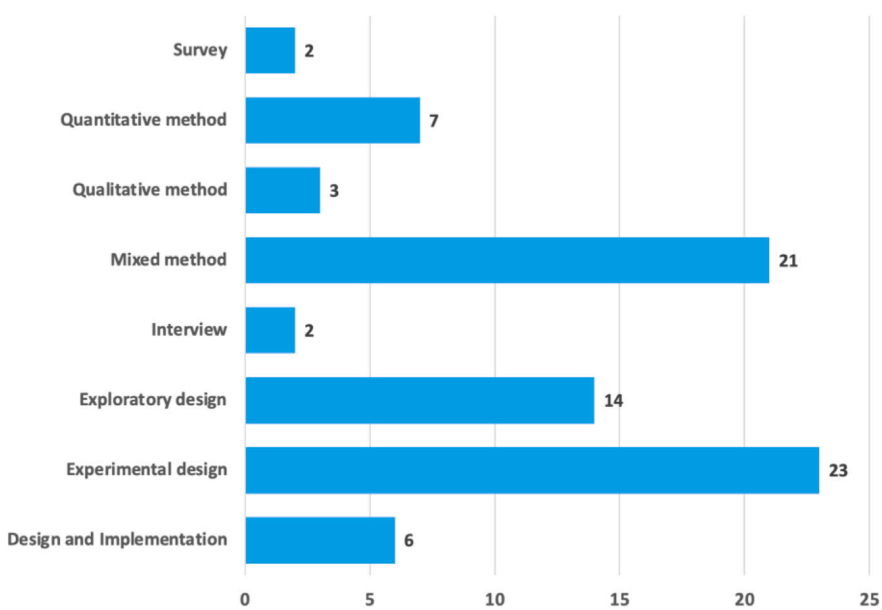

Fig. 5. Methodology.

which typically saves them more time to contextualize and refine the content. For example, there are studies that used ChaptGPT to generate multiple choice questions for an introductory programming class [66, 77]. In addition, the analysis shows that teachers are exploring how to use GenAI for automated assessment of programming questions [77], as teaching assistant [67], and in engaging with students to brainstorm and ideate conceptual designs [38]. With GenAI tools, one can generate codes that address a programming problem using multiple programming languages, which could enhance the mastery of other programming languages.

Furthermore, we examined the articles to uncover issues reported by scholars regarding the use of GenAI in computing education as shown in Table 7. We categorized these issues into 'mild' and 'severe' depending on how the issues were reported. The authors considered an issue mild if it was not reported to negatively significantly impact the performance of the GenAI tool based on the objective of the study. On the other hand, we consider an issue severe if there is indication that it caused a negative significant impact on what is being investigated in the study. For example, a study that investigated GenAI's capability of generating accurate and timely response and found that it could support formative

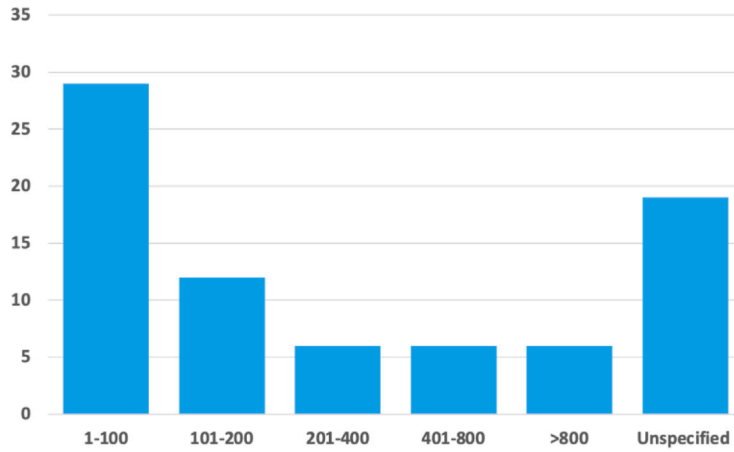

Fig. 6. Sample sizes.

Table 6
Potential benefits of GenAI in computing education reported in the literature.

The pros reported in the articles	Occurrences	Articles
Personalized learning experience	14	[16,19,22,43,45,47–54]
Code generation	12	[19,21,55–60]
Personalized feedback	7	[28,45,47,56,61–63]
Enhanced error messages	7	[28,46,61,64,65]
Developing useful teaching and learning resources	6	[36,54,66,67]
Clear error identification and fix suggestions	5	[28,64,68]
Personalized code explanation	5	[31,51,59,64,69]
Timely code support	4	[56,68,70]
Brainstorming and ideation	3	[38,56]
Better performance (GPT-4)	3	[42,55,71]
Automated assessment	3	[23,61,72]
Code debugging	1	[56]
Content creation	1	[56]
Engagements within problem context	1	[68]
Good Persona design and training	1	[36]
Improved accuracy	1	[73]
Good at solving Parson problems	1	[74]
Code interpretation	1	[70]
Timely teaching assistance	1	[67]
Multilingual programming	1	[75]
GenAI-human pair programming	1	[76]

feedback instead of summative, was deemed mild [78]. Similarly, if a study investigates GenAI's ability to review learner source code to identify errors and found that the tool could not significantly achieve that aim, it was deemed to be severe [79].

This categorization is helpful in understanding the nuances surrounding the integration of GenAI in computing classes so that educators can create innovative ways to achieve optimal solutions void of already reported issues. In general, the issue with prompt engineering was commonly reported among the articles as the LLM can only act on the input (descriptive text) given to it. If the user did not provide a coherent prompt, the LLM tool may not be able to generate an accurate response. Therefore prompt engineering is a requisite skill that users must have in order to maximize the benefits of GenAI tools in computing.

5. Discussion

In the light of the growing initiative to integrate GenAI tools into teaching and learning across educational levels, especially within the context of computing education, it is imperative to aggregate the existing research efforts for effective integration of GenAI into computing education and identify future research gaps. Unlike previous reviews that specifically focused on GenAI in programming education [2,13], this study investigates a wider scope of computing education to address two main research questions regarding how GenAI tools have been used to teach and learn in computer science education and what the pros and cons of integrating GenAI tools in computing education are. We then reflect on the result of the review for each of the two research questions guiding this study.

5.1. Exploring GenAI tools in computer science education

To understand how GenAI tools have been used in computing education, we explored educational level with which the studies are targeted, the topics in computing that were taught using the tools, the educational stakeholders using tools, the GenAI tools that were used, the methodological strategies used in exploring GenAI as well as the characteristics of the sample sizes in the selected articles.

GenAI's Impact on Teaching Practices and Curriculum Design

The systematic review highlights that educators are actively exploring GenAI tools such as ChatGPT-3/4 to innovate pedagogical strategies, particularly in introductory programming and guided learning interventions [2,3]. Although these tools are increasingly integrated into undergraduate curricula, their application in specialized computing domains (e.g., human–computer interaction, software engineering, and data science) remains underexplored [23,38]. Teachers predominantly adopt experimental or mixed methods to assess the efficacy of GenAI, although more in-depth investigations of its utility across diverse computing topics are critical to inform curriculum design [39, 40].

Table 7
Issues reported in the articles utilizing GenAI in computing education.

Mild issues reported (No of occurrence)	Articles	Severe issues reported (No of occurrence)	Articles
Formative rather than summative feedback (1)	[78]	Inability to review codes (1)	[79]
Limited in code error explanation (1)	[65]	Limited collaborations (1)	[76]
Performance not generalizable in all languages (1)	[75]	Limited in object-oriented design (1)	[70]
Inclusion among diverse demographics (2)	[75,78]	Perceived lack of programming fundamentals (1)	[80]
Needs instructor's supervision (2)	[68,77]	Poor in code generation for UX design (1)	[38]
Setting rules for students' engagement (2)	[43,53]	Poor output formatting (1)	[44]
Limited accuracy and performance (4)	[35,55,65,71]	Reliability concern (1)	[56]
Impacted with prompt engineering (4)	[73,74,81]	Repetition (1)	[36]
		Self-sabotage (1)	[35]
		Concern on suitability for novices (1)	[38]
		Bias, fairness, inclusion, and diversity concern (1)	[78]
		Ethical use (2)	[68,78]
		Lacks understanding of complex problems (2)	[57,67,82]
		Verbose (2)	[23,36]
		Poor code generation for file handling and complex problems (3)	[33,57,67]
		Domain concept understanding (3)	[40,83,84]
		Inconsistency in feedback generation (3)	[78,85]
		Limits learners' critical thinking (3)	[39,80]
		Occasionally generates incomplete or broken codes (3)	[59,70]
		Accuracy concern (4)	[49,56,68,86]
		Hallucinations/ Confabulation (4)	[85,87]
		Incorrect error explanations or suggestion for fix (4)	[65,86]
		Perceived fear of over-reliance by learners (4)	[33,35,40,80]
		Occasional refusal - (some test cases) (6)	[19,39,49,82,87,88]
		Prompt engineering skills required (9)	[19,36,54,55,81,82,84,89]

GenAI's Role in Student Learning Dynamics

Students engage primarily with GenAI through teacher-guided practices, with limited evidence of its effectiveness in fostering self-regulated learning [25]. While GenAI demonstrates potential to enhance engagement in programming education, its use in non-traditional settings (e.g., K-12, adult education, and non-formal environments) is significantly understudied [22]. Furthermore, research disproportionately focuses on undergraduate cohorts, leaving gaps in understanding how tools like GPT-4 or domain-specific platforms (e.g., [46]) support learners at other educational stages.

Institutional Adoption and Research Methodologies

Institutions face challenges in scaling GenAI integration, as most studies are based on small sample sizes (1 to 100 participants) and short-term interventions [28]. While experimental and mixed-method designs dominate current research, longitudinal studies are needed to evaluate sustained impacts on learning outcomes. The predominance of ChatGPT underscores a lack of comparative analyses with newer or specialized tools, limiting insights into their pedagogical affordances. Institutions must prioritize large-scale trials and cross-topic investigations to address ethical, equity, and efficacy concerns in computing education.

To advance GenAI's responsible adoption, collaborative efforts among educators, students and institutions should focus on (i) developing frameworks for the deployment of ethical tools in various computing domains; (ii) expanding research in K-12 and non-formal learning contexts; (iii) comparing GPT variants and domain-specific tools (for example, [45]) to identify optimal use cases; and (iv) addressing infrastructure and training gaps to support equitable access.

GenAI as a Pedagogical Catalyst: Bridging Theory and Practice

The findings in this study underscore GenAI's alignment with constructivist and cognitivist principles, bridging theoretical foundations with practical applications. By enabling iterative problem-solving (for example, code refinement using ChatGPT), GenAI tools fosters learner autonomy while reducing cognitive load through syntax automation [2,28]. However, over-reliance on GenAI may inadvertently marginalize social constructivist elements, a tension noted in studies prioritizing tool-driven guidance over collaborative learning [3,23]. These insights advocate for balanced integration, where GenAI complements rather than replaces—theories emphasizing social interaction and metacognitive growth.

5.2. The pros and cons of integrating GenAI tools in computing education

Here, we reflect on the findings about the potential benefits and challenges of using GenAI in computing education as reported in the selected articles. With respect to the potential benefits, 21 different learning gains were identified as shown in Table 6. Of all these learning gains, personalized learning experience emerged as the most prevalent benefit of GenAI for computing education in the studies. This finding is consistent with the study of Mosaiyebzadeh et al. [90] that explored the role of ChatGPT in education. In line with previous research on GenAI in computing education [2,13], code generation is a prominent activity that GenAI has been helpful in achieving in computing education. In addition, the chatbot application has been effective in improving students' performance in debugging through enhanced programming error messages. These findings are in line with earlier report that indicates programming education as the prevalent topic in computing that has been explored using GenAI [3].

The emergence of GenAI was accompanied with concerns about academic integrity [4] among other issues which still persist today. We classified the issues uncovered in the papers into two categories which we tagged mild and severe issues based on how they were reported. We identified 8 mild issues and 25 severe issues. As indicated earlier, the issues are considered mild if it does not negatively affect GenAI tool performance based on a study's objective and the issue is deemed severe if there is evidence that the tool caused a significant impact negatively on the subject under investigation. Most of the mild issues reported borders on limited accuracy and performance and issues with prompt engineering which are interrelated. Similarly, the requirement of prompt engineering skills top the list of the severe issues raised in the studies. This finding highlights the need to explore prompt engineering as a strategy to learn or resolve errors in programming. While studies such as Denny et al. [81] have begun to explore the efficacy of prompt engineering for Copilot in an introductory programming context, more work is required to examine its pedagogical value for learning computing using GenAI.

5.3. Study limitation

This study is not without limitations as common in review studies [3,30]. First, this study does not claim to be exhaustive but systematic. The research domain is still new and maturing, thus, it is possible to have left out several data in the process of data collection which limits the generalizability of the findings. Moreover, this study set out several exclusion criteria that may have introduced bias. For example, this study excluded articles that are not written in English language. Since GenAI is driven by LLM with multilingual capabilities, we imagine articles in diverse languages aside from English exists. In addition, we also excluded articles that were not peer-reviewed in order to report based on scientific evidence. Despite these limitations, this study provides an opportunity for future studies by validating the findings in this study through empirical studies.

6. Conclusions and future work

This paper investigates the use of GenAI in computing education research through the lens of a systematic literature review and contributes to understanding the landscape, including describing how GenAI tools have been used and its pros and cons. The aggregate of the 78 articles present the latest insight into how stakeholders have been exploring new tools engendered by GenAI in computing education domain.

Following the findings from this study, it will be important to provide perspective for future research focusing on addressing the challenges found and complementing the potential benefits of GenAI in computing education. Thus, we suggest the following for future work to further research agenda on GenAI for computing education:

1. In helping teachers to automate grading, GenAI has been found useful. Previous research has alluded to how cumbersome it is to scale grading for a large programming class [91]. Future study should explore computer science curriculum design with emphases on assessment of learning outcomes in the era of GenAI. This could foster computing education at scale where there is resource constraint.
2. Issues related to hallucination/confabulation still persists and more research could help guide future advancements.
3. Developing an instructional guide for using GenAI tools in computer science classroom should be a paramount future agenda. This will help in maintaining ethical use of GenAI tools by students and could mitigate the fear of over-reliance.
4. Both teachers and students require the skill of prompt engineering to meaningfully engage GenAI tools for productive teaching and learning outcomes. This skill is also required to reduce the "refusal" problem where the GenAI tool could not perform to a tasks or could not provide a response.
5. Broadening participation through the use of GenAI in computing education should be paramount. For instance, our search did not return papers reporting data from an Africa country or persons with disabilities.
6. To further prepare students for industry-aligned expertise where AI literacy is required, there is need to encourage the use of GenAI across subfields of computing education curricula, and at various levels. This effort will provide strong evidence based on literature in future reviews on GenAI in computing education.

As an evolving field, GenAI in education continues to generate interest. Within the computing education community, we envisage that this trend will continue. While this study reports on current scientific evidence, more effort is needed to shape the future of computing education to prepare students with relevant skills for future work.

This systematic review's findings underscore GenAI's evolving role across computing education disciplines, revealing domain-specific pedagogical needs, ethical consideration, integration challenges, and risks tied to impeding learning process or outcomes due to overreliance. By synthesizing evidence across subfields, the study equips computing educators with methodologies to design curricula that balance AI-assisted efficiency with the preservation of core competencies such as critical thinking, system design, and ethical reasoning. These insights empower stakeholders to adopt nuanced strategies that leverage GenAI's capabilities while safeguarding discipline-specific rigor, ensuring students develop adaptable, industry-aligned expertise in an AI-augmented professional landscape.

CRediT authorship contribution statement

Friday Joseph Agbo: Writing – review & editing, Writing – original draft, Supervision, Formal analysis, Data curation, Conceptualization. **Chris Olivia:** Writing – review & editing, Writing – original draft. **Godsalvation Oguibe:** Writing – review & editing, Writing – original draft. **Ismaila Temitayo Sanusi:** Writing – review & editing, Writing – original draft, Resources. **Godwin Sani:** Writing – review & editing, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Lo CK. What is the impact of ChatGPT on education? A rapid review of the literature. Educ Sci 2023;13(4):410.

[2] Prather J, Denny P, Leinonen J, Becker BA, Albluwi I, Craig M, Keuning H, Kiesler N, Kohn T, Luxton-Reilly A, et al. The robots are here: Navigating the generative ai revolution in computing education. In: Proceedings of the 2023 working group reports on innovation and technology in computer science education. 2023, p. 108–59.

[3] Cambaz D, Zhang X. Use of AI-driven code generation models in teaching and learning programming: a systematic literature review. In: SIGCSE 2024, New York, NY, USA: Association for Computing Machinery; 2024, p. 172–8. http://dx.doi.org/10.1145/3626252.3630958.

[4] Lee VR, Pope D, Miles S, Zarate R. Cheating in the age of generative AI: A high school survey study of cheating behaviors before and after the release of ChatGPT. Comput Educ: Artif Intell 2024;100253.

[5] Berry DM. The limits of computation: Joseph weizenbaum and the ELIZA chatbot. Weizenbaum J Digit Soc 2023;3(3).

[6] Bobrow D, et al. Natural language input for a computer problem solving system. Massachusetts Institute of Technology, MIT; 1964.

[7] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. Adv Neural Inf Process Syst 2014;27.

[8] Hinton G, Srivastava N, Swersky K. Neural networks for machine learning. Coursera, Video Lect 2012;264(1):2146–53.

[9] Lund BD, Wang T, Mannuru NR, Nie B, Shimray S, Wang Z. ChatGPT and a new academic reality: Artificial intelligence-written research papers and the ethics of the large language models in scholarly publishing. J Assoc Inf Sci Technol 2023;74(5):570–81.

[10] Motlagh NY, Khajavi M, Sharifi A, Ahmadi M. The impact of artificial intelligence on the evolution of digital education: A comparative study of openai text generation tools including ChatGPT, bing chat, bard, and ernie. 2023, arXiv preprint arXiv:2309.02029.

[11] Duong CD, Vu TN, Ngo TVN. Applying a modified technology acceptance model to explain higher education students' usage of ChatGPT: A serial multiple mediation model with knowledge sharing as a moderator. Int J Manag Educ 2023;21(3):100883.

[12] Hu B, Zheng L, Zhu J, Ding L, Wang Y, Gu X. Teaching plan generation and evaluation with GPT-4: Unleashing the potential of LLM in instructional design. IEEE Trans Learn Technol 2024.

[13] Deriba FG, Sanusi IT, Sunday AO. Enhancing computer programming education using ChatGPT-a mini review. In: Proceedings of the 23rd koli calling international conference on computing education research. 2023, p. 1–2.

[14] Zheng C, Yuan K, Guo B, Hadi Mogavi R, Peng Z, Ma S, Ma X. Charting the future of AI in project-based learning: A co-design exploration with students. In: Proceedings of the CHI conference on human factors in computing systems. 2024, p. 1–19.

[15] Tang Y, Chen L, Chen Z, Chen W, Cai Y, Du Y, Yang F, Sun L. EmoEden: Applying generative artificial intelligence to emotional learning for children with high-function autism. In: Proceedings of the CHI conference on human factors in computing systems. 2024, p. 1–20.
[16] Zheng Y. ChatGPT for teaching and learning: An experience from data science education. In: Proceedings of the 24th annual conference on information technology education. 2023, p. 66–72.
[17] Xia Q, Weng X, Ouyang F, Lin TJ, Chiu TK. A scoping review on how generative artificial intelligence transforms assessment in higher education. Int J Educ Technol High Educ 2024;21(1):40.
[18] Agbo FJ, Smalley HK, Nyman K. Competency-based assessment in the era of generative artificial intelligence: Perspectives of selected STEM educators. In: International conference on computers in education. 2024.
[19] Shen Y, Ai X, Soosai Raj AG, Leo John RJ, Syamkumar M. Implications of ChatGPT for data science education. In: Proceedings of the 55th ACM technical symposium on computer science education v. 1. 2024, p. 1230–6.
[20] Muller M, Chilton LB, Kantosalo A, Martin CP, Walsh G. Genaichi: generative AI and HCI. In: CHI conference on human factors in computing systems extended abstracts. 2022, p. 1–7.
[21] Kazemitabaar M, Chow J, Ma CKT, Ericson BJ, Weintrop D, Grossman T. Studying the effect of ai code generators on supporting novice learners in introductory programming. In: Proceedings of the 2023 CHI conference on human factors in computing systems. 2023, p. 1–23.
[22] Abolnejadian M, Alipour S, Taeb K. Leveraging ChatGPT for adaptive learning through personalized prompt-based instruction: A CS1 education case study. In: Extended abstracts of the CHI conference on human factors in computing systems. 2024, p. 1–8.
[23] Freire AP, Cardoso PCF, Salgado Ad. May we consult ChatGPT in our human-computer interaction written exam? an experience report after a professor answered yes. In: Proceedings of the XXII Brazilian symposium on human factors in computing systems. 2023, p. 1–11.
[24] Tan MJT, Maravilla NMAT. Shaping integrity: why generative artificial intelligence does not have to undermine education. Front Artif Intell 2024;7:1471224.
[25] Chiu TK. A classification tool to foster self-regulated learning with generative artificial intelligence by applying self-determination theory: A case of ChatGPT. Educ Technol Res Dev 2024;1–16.
[26] Anjali S, et al. The impact of artificial intelligence in reshaping education: An analysis based on learning theories. In: ITM web of conferences. 68, EDP Sciences; 2024, p. 01008.
[27] Wu F, Dang Y, Li M. A systematic review of responses, attitudes, and utilization behaviors on generative AI for teaching and learning in higher education. Behav Sci 2025;15(4):467.
[28] MacNeil S, Denny P, Tran A, Leinonen J, Bernstein S, Hellas A, Sarsa S, Kim J. Decoding logic errors: A comparative study on bug detection by students and large language models. In: Proceedings of the 26th australasian computing education conference. 2024, p. 11–8.
[29] Page MJ, McKenzie JE, Bossuyt PM, Boutron I, Hoffmann TC, Mulrow CD, Shamseer L, Tetzlaff JM, Akl EA, Brennan SE, et al. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. Bmj 2021;372.
[30] Agbo FJ. Broadening participation in adult education: A literature review of computer science education. In: Proceedings of the 55th ACM technical symposium on computer science education v. 1. 2024, p. 11–7.
[31] MacNeil S, Tran A, Mogil D, Bernstein S, Ross E, Huang Z. Generating diverse code explanations using the gpt-3 large language model. In: Proceedings of the 2022 ACM conference on international computing education research-volume 2. 2022, p. 37–9.
[32] Sarsa S, Denny P, Hellas A, Leinonen J. Automatic generation of programming exercises and code explanations using large language models. In: Proceedings of the 2022 ACM conference on international computing education research-volume 1. 2022, p. 27–43.
[33] Finnie-Ansley J, Denny P, Becker BA, Luxton-Reilly A, Prather J. The robots are coming: Exploring the implications of openai codex on introductory programming. In: Proceedings of the 24th australasian computing education conference. 2022, p. 10–9.
[34] Vaithilingam P, Zhang T, Glassman EL. Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In: Chi conference on human factors in computing systems extended abstracts. 2022, p. 1–7.
[35] Joshi I, Budhiraja R, Dev H, Kadia J, Ataullah MO, Mitra S, Akolekar HD, Kumar D. ChatGPT in the classroom: An analysis of its strengths and weaknesses for solving undergraduate computer science questions. In: Proceedings of the 55th ACM technical symposium on computer science education v. 1. 2024, p. 625–31.
[36] Barambones J, Moral C, de Antonio A, Imbert R, Martínez L, Villalba-Mora E. ChatGPT for learning HCI techniques: A case study on interviews for personas. IEEE Trans Learn Technol 2024.
[37] Dobslaw F, Bergh P. Experiences with remote examination formats in light of GPT-4. In: Proceedings of the 5th European conference on software engineering education. 2023, p. 220–5.
[38] York E. Evaluating ChatGPT: Generative AI in UX design and web development pedagogy. In: Proceedings of the 41st ACM international conference on design of communication. 2023, p. 197–201.
[39] Feng TH, Denny P, Wuensche B, Luxton-Reilly A, Hooper S. More than meets the AI: Evaluating the performance of GPT-4 on computer graphics assessment questions. In: Proceedings of the 26th australasian computing education conference. 2024, p. 182–91.
[40] Petrovska O, Clift L, Moller F, Pearsall R. Incorporating generative AI into software development education. In: Proceedings of the 8th conference on computing education practice. 2024, p. 37–40.
[41] Savelka J, Agarwal A, Bogart C, Song Y, Sakr M. Can generative pre-trained transformers (gpt) pass assessments in higher education programming courses? In: Proceedings of the 2023 conference on innovation and technology in computer science education v. 1. 2023, p. 117–23.
[42] Jin H, Lee S, Shin H, Kim J. Teach AI how to code: Using large language models as teachable agents for programming education. In: Proceedings of the CHI conference on human factors in computing systems. 2024, p. 1–28.
[43] Bhalerao R. My learnings from allowing large language models in introductory computer science classes. In: Proceedings of the 55th ACM technical symposium on computer science education v. 2. 2024, p. 1574–5.
[44] Hellas A, Leinonen J, Sarsa S, Koutcheme C, Kujanpää L, Sorva J. Exploring the responses of large language models to beginner programmers' help requests. In: Proceedings of the 2023 ACM conference on international computing education research-volume 1. 2023, p. 93–105.
[45] Liu R, Zenke C, Liu C, Holmes A, Thornton P, Malan DJ. Teaching CS50 with AI: leveraging generative artificial intelligence in computer science education. In: Proceedings of the 55th ACM technical symposium on computer science education v. 1. 2024, p. 750–6.
[46] Kuramitsu K, Obara Y, Sato M, Obara M. KOGI: A seamless integration of ChatGPT into jupyter environments for programming education. In: Proceedings of the 2023 ACM SIGPLAN international symposium on SPLASH-e. 2023, p. 50–9.
[47] Azaiz I, Deckarm O, Strickroth S. AI-enhanced auto-correction of programming exercises: How effective is GPT-3.5?. 2023, arXiv preprint arXiv:2311.10737.
[48] Glynn C, Hed E, Pexa A, Pohlmann T, Rahal I, Hesse R. CAET: Code analysis and education tutor. In: Proceedings of the 55th ACM technical symposium on computer science education v. 2. 2024, p. 1656–7.
[49] Qureshi B. ChatGPT in computer science curriculum assessment: An analysis of its successes and shortcomings. In: Proceedings of the 2023 9th international conference on e-society, e-learning and e-technologies. 2023, p. 7–13.
[50] Liffiton M, Sheese BE, Savelka J, Denny P. Codehelp: Using large language models with guardrails for scalable support in programming classes. In: Proceedings of the 23rd koli calling international conference on computing education research. 2023, p. 1–11.
[51] Jury B, Lorusso A, Leinonen J, Denny P, Luxton-Reilly A. Evaluating LLM-generated worked examples in an introductory programming course. In: Proceedings of the 26th australasian computing education conference. 2024, p. 77–86.
[52] MacNeil S, Tran A, Hellas A, Kim J, Sarsa S, Denny P, Bernstein S, Leinonen J. Experiences from using code explanations generated by large language models in a web software development e-book. In: Proceedings of the 54th ACM technical symposium on computer science education v. 1. 2023, p. 931–7.
[53] Zastudil C, Rogalska M, Kapp C, Vaughn J, MacNeil S. Generative ai in computing education: Perspectives of students and instructors. In: 2023 IEEE frontiers in education conference. IEEE; 2023, p. 1–9.
[54] Nguyen S, Babe HM, Zi Y, Guha A, Anderson CJ, Feldman MQ. How beginning programmers and code LLMs (mis) read each other. In: Proceedings of the CHI conference on human factors in computing systems. 2024, p. 1–26.
[55] Finnie-Ansley J, Denny P, Luxton-Reilly A, Santos EA, Prather J, Becker BA. My ai wants to know if this will be on the exam: Testing openai's codex on cs2 programming exercises. In: Proceedings of the 25th australasian computing education conference. 2023, p. 97–104.
[56] Budhiraja R, Joshi I, Challa JS, Akolekar HD, Kumar D. "It's not like jarvis, but it's pretty close!"-examining ChatGPT's usage among undergraduate students in computer science. In: Proceedings of the 26th australasian computing education conference. 2024, p. 124–33.
[57] Sharpe JS, Dougherty RE, Smith SJ. Can ChatGPT pass a CS1 python course? J Comput Sci Coll 2024;39(8):128–42.
[58] Xu ZT, Xu RT, Sheng VS. ChatGPT-generated code assignment detection using perplexity of large language models. 2024.
[59] Taylor A, Vassar A, Renzella J, Pearce H. Dcc–help: Transforming the role of the compiler by generating context-aware error explanations with large language models. In: Proceedings of the 55th ACM technical symposium on computer science education v. 1. 2024, p. 1314–20.
[60] Xu Z, Sheng VS. Detecting AI-generated code assignments using perplexity of large language models. In: Proceedings of the AAAI conference on artificial intelligence. 38, (21):2024, p. 23155–62.
[61] Kimmel B, Geisert AL, Yaro L, Gipson B, Hotchkiss RT, Osae-Asante SK, Vaught H, Wininger G, Yamaguchi C. Enhancing programming error messages in real time with generative AI. In: Extended abstracts of the CHI conference on human factors in computing systems. 2024, p. 1–7.

[62] Cowan B, Watanobe Y, Shirafuji A. Enhancing programming learning with LLMs: Prompt engineering and flipped interaction. In: Proceedings of the 2023 4th Asia service sciences and software engineering conference. 2023, p. 10–6.
[63] Nguyen H, Allan V. Using GPT-4 to provide tiered, formative code feedback. In: Proceedings of the 55th ACM technical symposium on computer science education v. 1. 2024, p. 958–64.
[64] Santos EA, Prasad P, Becker BA. Always provide context: The effects of code context on programming error message enhancement. In: Proceedings of the ACM conference on global computing education vol 1. 2023, p. 147–53.
[65] Balse R, Kumar V, Prasad P, Warriem JM. Evaluating the quality of LLM-generated explanations for logical errors in CS1 student programs. In: Proceedings of the 16th annual ACM India compute conference. 2023, p. 49–54.
[66] Doughty J, Wan Z, Bompelli A, Qayum J, Wang T, Zhang J, Zheng Y, Doyle A, Sridhar P, Agarwal A, et al. A comparative study of AI-generated (GPT-4) and human-crafted MCQs in programming education. In: Proceedings of the 26th australasian computing education conference. 2024, p. 114–23.
[67] Laney M, Dewan P. Human-AI collaboration in a student discussion forum. In: Companion proceedings of the 29th international conference on intelligent user interfaces. 2024, p. 74–7.
[68] Liu M, M'hiri F. Beyond traditional teaching: Large language models as simulated teaching assistants in computer science. In: Proceedings of the 55th ACM technical symposium on computer science education v. 1. 2024, p. 743–9.
[69] Leinonen J, Denny P, MacNeil S, Sarsa S, Bernstein S, Kim J, Tran A, Hellas A. Comparing code explanations created by students and large language models. In: Proceedings of the 2023 conference on innovation and technology in computer science education v. 1. 2023, p. 124–30.
[70] Cipriano BP, Alves P. Gpt-3 vs object oriented programming assignments: An experience report. In: Proceedings of the 2023 conference on innovation and technology in computer science education v. 1. 2023, p. 61–7.
[71] Poulsen S, Sarsa S, Prather J, Leinonen J, Becker BA, Hellas A, Denny P, Reeves BN. Solving proof block problems using large language models. In: Proceedings of the 55th ACM technical symposium on computer science education v. 1. 2024, p. 1063–9.
[72] Xiang L. SQL query evaluation with large language model and abstract syntax trees. In: Proceedings of the 55th ACM technical symposium on computer science education v. 2. 2024, p. 1890–1890.
[73] Venkatesh V, Venkatesh V, Kumar V. Evaluating copilot on CS1 code writing problems with suppressed specifications. In: Proceedings of the 16th annual ACM India compute conference. 2023, p. 104–7.
[74] Reeves B, Sarsa S, Prather J, Denny P, Becker BA, Hellas A, Kimmel B, Powell G, Leinonen J. Evaluating the performance of code generation models for solving parsons problems with small prompt variations. In: Proceedings of the 2023 conference on innovation and technology in computer science education v. 1. 2023, p. 299–305.
[75] Jordan M, Ly K, Soosai Raj AG. Need a programming exercise generated in your native language? ChatGPT's got your back: Automatic generation of non-english programming exercises using openai GPT-3.5. In: Proceedings of the 55th ACM technical symposium on computer science education v. 1. 2024, p. 618–24.
[76] Chen X, Liang J. Pair programming with ChatGPT. In: Proceedings of the 55th ACM technical symposium on computer science education v. 2. 2024, p. 1600–1.
[77] Tran A, Angelikas K, Rama E, Okechukwu C, Smith DH, MacNeil S. Generating multiple choice questions for computing courses using large language models. In: 2023 IEEE frontiers in education conference. IEEE; 2023, p. 1–8.
[78] Woodrow J, Malik A, Piech C. Ai teaches the art of elegant coding: Timely, fair, and helpful style feedback in a global course. In: Proceedings of the 55th ACM technical symposium on computer science education v. 1. 2024, p. 1442–8.
[79] Pirttinen N, Leinonen J. Could ChatGPT be used for reviewing learnersourced exercises? In: Proceedings of the 23rd koli calling international conference on computing education research. 2023, p. 1–2.
[80] Sheard J, Denny P, Hellas A, Leinonen J, Malmi L, Simon. Instructor perceptions of AI code generation tools-a multi-institutional interview study. In: Proceedings of the 55th ACM technical symposium on computer science education v. 1. 2024, p. 1223–9.
[81] Denny P, Kumar V, Giacaman N. Conversing with copilot: Exploring prompt engineering for solving cs1 problems using natural language. In: Proceedings of the 54th ACM technical symposium on computer science education v. 1. 2023, p. 1136–42.
[82] Kazemitabaar M, Hou X, Henley A, Ericson BJ, Weintrop D, Grossman T. How novices use LLM-based code generators to solve CS1 coding tasks in a self-paced learning environment. In: Proceedings of the 23rd koli calling international conference on computing education research. 2023, p. 1–12.
[83] Mahon J, Mac Namee B, Becker BA. No more pencils no more books: Capabilities of generative AI on irish and UK computer science school leaving examinations. In: Proceedings of the 2023 conference on United kingdom & Ireland computing education research. 2023, p. 1–7.
[84] Prasad S, Greenman B, Nelson T, Krishnamurthi S. Generating programs trivially: Student use of large language models. In: Proceedings of the ACM conference on global computing education vol 1. 2023, p. 126–32.
[85] Balse R, Valaboju B, Singhal S, Warriem JM, Prasad P. Investigating the potential of gpt-3 in providing feedback for programming assessments. In: Proceedings of the 2023 conference on innovation and technology in computer science education v. 1. 2023, p. 292–8.
[86] Leinonen J, Hellas A, Sarsa S, Reeves B, Denny P, Prather J, Becker BA. Using large language models to enhance programming error messages. In: Proceedings of the 54th ACM technical symposium on computer science education v. 1. 2023, p. 563–9.
[87] Hou I, Man O, Mettille S, Gutierrez S, Angelikas K, MacNeil S. More robots are coming: Large multimodal models (ChatGPT) can solve visually diverse images of parsons problems. In: Proceedings of the 26th australasian computing education conference. 2024, p. 29–38.
[88] Savelka J, Agarwal A, An M, Bogart C, Sakr M. Thrilled by your progress! large language models (GPT-4) no longer struggle to pass assessments in higher education programming courses. In: Proceedings of the 2023 ACM conference on international computing education research-volume 1. 2023, p. 78–92.
[89] Bellettini C, Lodi M, Lonati V, Monga M, Morpurgo A. Davinci goes to bebras: a study on the problem solving ability of GPT-3. In: CSEDU 2023-15th international conference on computer supported education. 2, SCITEPRESS-Science and Technology Publications; 2023, p. 59–69.
[90] Mosaiyebzadeh F, Pouriyeh S, Parizi R, Dehbozorgi N, Dorodchi M, Macêdo Batista D. Exploring the role of ChatGPT in education: applications and challenges. In: Proceedings of the 24th annual conference on information technology education. 2023, p. 84–9.
[91] Messer M, Brown NC, Kölling M, Shi M. Automated grading and feedback tools for programming education: A systematic review. ACM Trans Comput Educ 2024;24(1):1–43.