Pair Programming with AI: A Systematic Literature Review

Julie M. Smith[1]

[1]Institute for Advancing Computing Education
[1]julie@csedresearch.org

Abstract

Introduction In pair programming, two people work together on a programming task, with one serving as the "driver" and the other as the "navigator." Research on pair programming has been somewhat mixed: in many cases it is an effective approach for promoting positive learning outcomes, but it may not be beneficial to all students. Generative AI has created a new possibility for pair programming: pairs composed of one student and an AI tool. Despite its novelty, there is a growing body of research on AI pair programming.

Objective The objective of this study is to answer the research question: *What evidence exists regarding the outcomes of student-AI pair programming?*

Methods We queried five traditional collections of academic research; we also prompted two AI tools designed to identify academic research. We identified 10 relevant studies. We analyzed each to determine the context (e.g., grade level), the AI tool(s) used, the research type and design, what (if any) preparation was provided to participants, what outcomes were considered, and the study's findings.

Results Almost all studies involved college students. The studies used a variety of methods – from analysis of interview transcripts to surveys for affective constructs. The most commonly used AI tool was ChatGPT or a variant of it. Study findings varied, but there was evidence of positive learning outcomes paired with less social connection when AI tools were used. Other affective outcomes – such as engagement – could be enhanced or reduced when students were paired with an AI tool. Some findings indicated a preference for a human partner.

Discussion More research is needed to support these findings and to extend them to other contexts. Specifically, it is important to conduct research that explores how the factors important in traditional pair programming impact AI pair programming. Capturing the cognitive advantages that AI tools might offer students while mitigating the potential social harms will be an important part of the responsible development of AI pair programming as a pedagogical approach. We also comment on the performance of the various databases and AI tools used to locate relevant research.

AI Use Disclosure As noted above, AI was used to locate articles for the systematic literature review.

1 Introduction

Pedagogies, practices, and strategies for teaching computer science (CS) education have been evolving over the decades, with various levels of success. One such practice is pair programming, a collaborative learning experience for students in which two students work together to write a program. The practice involves one student who acts as the driver and is responsible for actually writing the code; the other is the navigator, assigned the role of observing, evaluating, and providing feedback as the work progresses. At intermittent times, the instructor will ask the students to switch roles, so that each is provided the opportunity to drive and to navigate.

Pair programming has been researched extensively in educational contexts, with multiple systematic literature reviews summarizing the findings [1–11]. In general, pair programming appears to benefit the process of learning to program and may also improve students' confidence [1], increase retention rates [3], and lead to increased enjoyment of programming [4]. However, some studies identify situations where pair programming may be less effective, and these are often linked to the composition characteristics of the partnership. For example, when a less experienced partner is paired with someone with more experience, the less experienced student exhibited less effort, less conceptual understanding, and less interest in computing, although there were not significant differences in grades [12]. Similarly, women students have reported that having a man as a partner resulted in behaviors indicating a lack of respect for their participation [13].

Given the increased use of generative AI in educational settings, there is an opportunity to explore AI within pair programming to potentially use AI to create a parallel pair programming experience. AI pair programming presents an opportunity to harness the advantages of both traditional pair programming and of AI use – if the potentially negative outcomes of some instantiations of pair programming and of AI use can be avoided. Research suggests that generative AI can create explanations of code that are easier to understand than those created by students [14], pointing to the possible benefits of AI pair programming. And human-AI pairs are likely to lack the affective and emotional aspects that human-human pairs have – whether this is to the detriment or to the benefit of students may depend on the context [15]. However, due to the recency of the widespread availability of the technology needed to support AI pair programming, little empirical evidence exists exploring the practice, particularly in educational settings.

Given the advent of AI pair programming, our research question for our systematic literature review (SLR) was:

> *What evidence exists regarding the outcomes of student-AI pair programming?*

While there are two closely-related literature reviews [16, 17] (see below), to our knowledge, there is no systematic literature review (SLR) focused specifically on AI pair programming in education.

2 Background

2.1 Generative AI to Support Programming

More than 90% of professional software developers already use AI to assist with their work [18], suggesting the utility of AI for programming. However, one study found that developers expected AI to improve their productivity by 20% when it actually decreased their productivity by a similar amount [19]. These findings suggest the combined promise and peril of the use of generative AI to assist with writing programs.

In educational contexts, productivity is not the key concern; learning is. And the use of generative AI introduces substantial concerns – as well as potentially important benefits – for the learning process. For example, when students completed similar learning tasks using either an AI tool or a standard web search, their knowledge gains were less when using the AI tool [20]. Similarly, students using AI tools may reduce their metacognitive monitoring and then become overly reliant on such tools to the detriment of their learning [21, 22]. Students may be confused by AI output, either because it uses unfamiliar programming constructs or because its output is wrong [23]. Further, when users have too much trust in AI, they may use the tools in unproductive ways, such as by relying on output that they lack the capacity to appropriately vet [24].

Even when AI is a boon to experienced programmers, it might be detrimental to novices, who lack the experience and knowledge to evaluate its output for correctness [25]. While access to AI might improve learning in the short-term, it may – once removed – result in worse performance relative to students who never used the tool [26], although carefully designed AI tutoring systems may mitigate this effect. On the other hand, some research has identified positive outcomes from AI-assisted learning [27, 28]. For example, students who used AI to help with writing or modifying code had better performance in the short term, with no negative impact in the longer term [29].

2.2 AI Pair Programming

In [16], AI-assisted programming in education was explored, and five research trends were identified: explorations of tools and technology, performance evaluation, impacts, behavior patterns, and ethical issues. Additionally, three suggestions for future research were made based on the reviewed literature: (1) acknowledging the capabilities as well as the limitations of AI, (2) developing appropriate educational models and conducting research to support AI use, and (3) exploring less traditional research methods. However, with an end date of November 2023 for publications, [16] does not capture the most recent research in this quickly-evolving field. Further, it focused on AI-assisted code generation literature generally (e.g., discussions of academic integrity concerns), not just studies that approximated traditional pair programming with AI. Also, it did not systematically track each study's context (e.g., students' grade level, programming language used, students' preparation to use AI if any).

In the other review, [17], the state of research related to human-AI pair programming was explored. The authors conclude that some of the challenges of human-human pairs (e.g., scheduling, assessing each person's contribution) may be mitigated by human-AI pair programming. The review concludes with suggestions for future research for supporting students in human-AI pair programming, including whether the AI's expertise should be leveled to the student's, how to prevent the AI from providing too much help, and whether the self-confidence gains of traditional pair programming (particularly for students with lower self-confidence) also occur with AI pair programming.

Unlike [16], [17] is limited to human-AI pair programming that is analogous to traditional pair programming, although it is not limited to educational contexts. This review makes no claim to being systematic, and no methodology for locating or including or excluding literature is included in the paper. As a 2023 publication, some of its observations no longer hold. For example, the authors note that "no current study sets up a three-way comparison for human-AI, human-human,

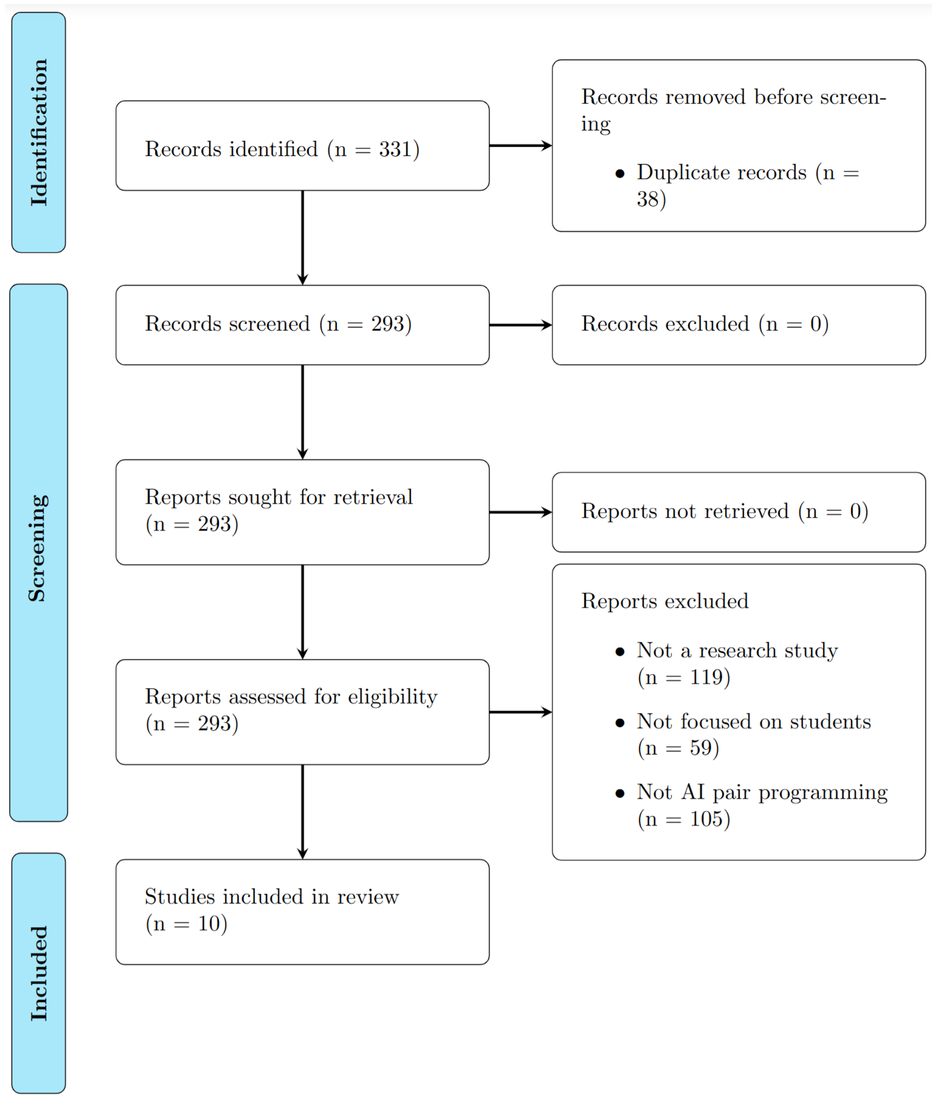

Figure 1: PRISMA Diagram

and human-solo" programming conditions [17], but see [30], [31], and [32]. Thus, there is a need for an SLR focused specifically on AI pair programming in education, with attention to the learning context.

3 Methodology

Using the Khan et al. methodology for conducting a SLR [33], we searched five traditional databases of academic research literature and used two AI tools designed to locate relevant research. See Figure 1, which summarizes the process of locating studies related to AI pair programming.

The five databases searched were Science Direct, ACM Digital Library, IEEE Xplore, Google Scholar, and the ASEE Paper Repository. For all save the ASEE Paper Repository, we used the search string *"pair programming" AND ("AI" or "ChatGPT" or "artificial intelligence")*. For the ASEE Paper Repository, we used a different approach due to the limitations of the search

Table 1: Literature Searches

Name	Type	Date	Total Results	Included Results
Science Direct	database	30 Sept 2025	20	0
ACM Digital Library	database	30 Sept 2025	150	0
IEEE Xplore	database	30 Sept 2025	8	2
Google Scholar	database	1 Oct 2025	30	7
ASEE Paper Repository	database	2 Oct 2025	74	0
Elicit	AI tool	1 Oct 2025	20	4
Asta	AI tool	1 Oct 2025	29	3

feature; we performed separate searches as follows: (1) "pair programming" ChatGPT, (2) "pair programming" AI, and (3) "pair programming" "artificial intelligence." This approach to searching led to some duplicates, which are included in the counts in Table 1 above. Google Scholar returned over 1,000 results, of which only the first 30 were further assessed due to the increasing prevalence of results that did not meet all of the search criteria (e.g., papers related to AI-supported programming but not *pair* programming with AI, papers related to pair programming but not *AI* pair programming).

The two AI tools used were Elicit (elicit.com) and Asta (asta.allen.ai). For both, the prompt "Please provide research studies that explore pair programming where an AI tool (such as ChatGPT) is half of the pair" was used. Elicit repeatedly offers a "load more" (i.e., more results) option; only the first 20 items were further assessed due to the increasing prevalence of results that did not meet all of the search criteria. Asta located "5 papers that look like perfect matches, 24 relevant ones[,] and 16 others;" the 16 'others' were ignored. Table 1 provides details about these searches.

This study had three criteria for inclusion:

- Content type: include research studies and exclude other content (e.g., posters, position papers)
- Context: include educational contexts (regardless of level) and exclude if solely other contexts (e.g., professional software developers)
- Focus: include if the focus of the design or intervention is AI pair programming and exclude other foci (e.g., AI tutors, traditional pair programming)

Note in regards to the final criterion, a focus on pair programming, that not all uses of AI while programming constitute AI pair programming. For example, two papers, "Agriculture students' use of generative artificial intelligence for microcontroller programming" [34] and "Enhancing programming pair workshops: The case of teacher pre-prompting" [35] were excluded from the dataset because, while student pairs used AI to write programs, they did not follow any of the key characteristics of pair programming with a human-AI pair, such as alternating roles as driver and navigator with the AI.

Titles and abstracts were reviewed against the inclusion criteria; as needed, the full text of the paper was consulted to determine whether all inclusion criteria were met.

One limitation of this study is that some papers that met our inclusion criteria were not included in the results of any of our searches. For example, [36] describes a quasi-experiment that assigned high school students to either traditional pair programming or to pair programming with ChatGPT. It is not clear why this study was not captured by any of seven searching platforms that we used.

4 Results and Discussion

4.1 Findings

We organized the results of the SLR and the ensuing discussion into four sections: a general overview of the findings, findings related to the student experience of AI pair programming, design considerations for AI pair programming, and the literature search process itself.

4.1.1 Overview of Findings

Table 2 summarizes the studies. While a total of 10 studies is perhaps not unexpected given the recency of modern generative AI technologies, it nonetheless illustrates the need for more research on AI pair programming. Of the 10 included studies, one did not specify the age or grade level of student participants [37] and one involved high school students [39]; the rest involve university students. Because traditional pair programming is used in classrooms with students as young as first grade [44], additional research on AI pair programming with K-12 students is warranted in order to better understand the impact of AI pair programming with elementary and secondary students as well as post-secondary students.

Studies included in the dataset had between 5 and 234 participants, for a total across all included studies of 532 participants and an average of 59 per study. These are relatively low numbers that, once again, point to the need for more research to ensure that the impact of AI pair programming across various contexts (e.g., levels of prior programming experience, personality traits such as introversion) and various implementations (e.g., remote pair programming, larger versus smaller assignments) are better understood. The need for more research with larger samples was noted also by [17].

Programming languages used by students in these studies were Python, Java, or HTML/CSS/JavaScript. Most studies used (or suggested that students use) ChatGPT or related AI tools; a few used Copilot or Claude. Once again, more research could expand this list of programming languages and AI tools. In particular, more AI tools should be studied (e.g., Gemini, Perplexity, Llama) to determine whether their performance as pair programming partners differs.

The set of papers in this study include a variety of study types (e.g., quasi-experimental, case study, observational). Interventions included comparisons between student-student and student-AI pairs and, sometimes, a solo programming condition. Some research was design-oriented in nature, sometimes with prototyping of AI tools. For example, in [38], participants believed that they were interacting with AI, but there was in fact a hybrid, Wizard-of-Oz design such that some of the “AI”’s actions were generated by the researcher. For such a small body of research, these study types and interventions represent a relatively large variety of approaches to the topic.

Table 2: Summary of Included Studies

Citation	Title	Context	AI Tool	Description	Intervention	AI Prep	Findings
[37]	Challenging the confirmation bias: Using ChatGPT as a virtual peer for peer instruction in computer programming education	CS classes; level/age not specified; n = 160	ChatGPT	quasi-experimental, mixed methods	student-student pairs or student-AI pairs	not clear	Students using ChatGPT showed better programming skills as well as more engagement and satisfaction. Further, teacher mediation was important.
[38]	From tool to partner: Exploring the roles of embodiment on AI agent in pair programming	University students; n=18, of which 12 had no prior programming experience	GPT-4o	repeated measures, mixed methods	in VR, embodied AI or voice-only AI	none	Embodied AI led to better engagement and communication, as well as sometimes changing trust in the AI. Programming learning was not significantly different.
[39]	Pair programming education aided by ChatGPT	High school students; n=45; Python	ChatGPT	pre- and post-test, quantitative	pair programming with AI	none	Students' programming interest increased, as did the belief that they learn faster with AI. In the post-test, students were more likely to think that they didn't need to learn to program.
[40]	Pair programming in programming courses in the era of generative AI: students' perspective	University students; n=12	ChatGPT and Copilot (suggested)	repeated measures, qualitative	student-student pairs or student-AI pairs	none	Students had less engagement and empathy when paired with AI; they preferred peer pairs.
[41]	Problem-solving through pair-programming: The mediational role of ChatGPT	University students; n=12; Python	ChatGPT	observational, qualitative	student-AI pairs	told not to request the entire program	Students consulted the AI for help with a variety of computational thinking concepts.
[30]	The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: a comparative study with traditional pair programming and individual approaches	University students; n=234; Java	GPT-3.5 Turbo and Claude 3 Opus	quasi-experiment	student-student pairs or student-AI pairs or solo work	model-specific best practices for educational AI pair programming	Student-AI pairs performed best on programming tasks, but student-student pairs had higher sense of collaboration and social presence.
[31]	The psychological effects of AI-assisted programming on students and professionals	University students; n=5	ChatGPT and Copilot	qualitative	solo, pair, and AI programming	pilot training session	Students felt human partners were better, even if AI partners were more efficient; introverts preferred AI over peer or solo.
[42]	Toward AI-facilitated learning cycle in integration course through pair programming with AI agents	University students	ChatGPT 4	case study (without actual students)	the AI was prompted to generative supportive materials for students	none	AI output was appropriate to support student learning.
[43]	Towards designing conversational agents for pair programming: Accounting for creativity strategies and conversational styles	University students; n=7; Java	N/A	observational, qualitative	students were paired with an AI agent in a Wizard-of-Oz study to complete a programming task	none	Several design guidelines for creativity and communication were difficult to implement in an AI.
[32]	Will your next pair programming partner be human? An empirical evaluation of generative AI as a collaborative teammate in a semester-long classroom setting	University students; n=39; web development	ChatGPT and Copilot	repeated measures, mixed methods	students completed assignments as student-student pairs, student-AI pairs, and solo with AI	tutorial for AI workflows	Students earned the highest scores when in student-AI pairs and the lowest scores when solo with AI; students developed an awareness of AI limitations.

We noted whether and how researchers addressed the fact that, unlike most human peers, an AI tool has vastly more knowledge about programming than the student. This topic was not addressed in most studies, but [41] notes that the disparity was mitigated by asking students not to ask the AI for the entire program but instead to work collaboratively with the tool. By contrast, in [42], the expert knowledge of the AI is treated as a given, and the student becomes the novice 'driver' while the AI is the expert 'navigator,' with the authors arguing that this structure permits the navigator to assume a mentoring role and leads to greater productivity. Similarly, [32] discusses qualitative findings suggesting that pairing with an LLM provided benefits to inexperienced students. In contrast, [41] notes that the AI tool introduced errors that should be mitigated by the students' knowledge and/or by careful prompting, although they note that students may lack this knowledge; inaccuracies were also noted by [40].

We also note that [43] suggests that AI tools may lack the social intelligence displayed by human-human pairs unless the tool is carefully designed to include it (e.g., detecting confusion or lack of engagement). In sum, differences in programming skill and in social skill between students and AI suggest that there may be differences between human-human and human-AI pair programming. More research will clarify the impacts of these differences and whether steps should be taken to mitigate the disparities or whether they instead contribute positively to the student experience.

We also noted whether students were given any particular preparation for AI pair programming. ([39] mentions that the instructor told students that ChatGPT did not indicate that it needed more information or did not know the answer to a question. Because this incident is described very briefly in the paper, it is not clear whether this observation occurred before the intervention [in which case it would be AI prep], during the intervention as an anecdotal remark, or after as a debriefing. Hence, we categorize it as a possible instance of AI prep.) For most studies, preparation not was mentioned, although some included direction not to request the entire solution from the AI, models of best practices for educational AI pair programming, or a training session. Future research should include various preparatory measures in order to assess their impact on the AI pair programming experience.

In general, study findings were mixed, with some studies showing that AI pair programming led to improvements in programming skills, engagement, satisfaction, communication, interest in programming, and perceptions of the speed of learning. However, other findings include no difference in learning to program and reductions in engagement, empathy, sense of collaboration, or social presence when students partnered with AI. Some findings indicated a preference for a human partner.

More research is needed to support these findings about affective factors and to extend them to other contexts. Specifically, it is important to conduct research that explores whether and how the factors important in traditional pair programming influence AI pair programming. For example, research on traditional pair programming suggests that students with less CS experience may have worse outcomes when paired with students with more experience. An unmodified AI tool will produce code akin to that of a more experienced student; thus, research that explores whether a restricted AI tool is a more effective pair programming partner might lead to important insights. Further, it is clear that capturing the cognitive advantages that AI tools might offer students while mitigating the potential social harms will be an important part of the responsible development of

AI pair programming as a pedagogical approach.

4.1.2 Student Experience Findings

The studies also reported more nuanced details about the experience of AI pair programming. Some of these details shed more light on the student experience of AI pair programming. [40] identified many use cases and challenges of AI pair programming: use cases included generating examples and debugging, and challenges included overly complicated and inaccurate solutions. [39] found that, after students had programmed with AI, they were more likely to agree that they used AI to create a complete solution, suggesting that they may not have avoided some of the dangers of over-reliance on AI tools. Participants also had greater agreement with the sentiment that they did not need to learn how to program, pointing to the potential for decreased interest in pursuing further studies in CS.

Perceptions of AI usefulness were found by [30] to be an important mediator to their findings, with positive impressions of AI linked to improved learning outcomes. The authors also suggest that their finding about reduced anxiety imply that AI pair programming may be particularly appropriate for students who struggle with anxiety. [31] explored attribution bias related to AI-assisted coding, finding that students sometimes overly attributed AI output to themselves, although students also thought the impact of AI on their learning was mixed. [32] found that attitudes toward AI pair programming became more positive over time, with students simultaneously becoming more aware of the limits of AI as a coding partner. On a more pragmatic level, students also expressed that AI pair programming was impeded when the AI tool was not part of their code editor (i.e., was located in a separate application).

4.1.3 Design Considerations Findings

Other details reported in the included papers suggest design considerations for future implementations of AI pair programming. [37] found that the instructor needed to mediate due to AI errors and/or to students not understanding the AI output, a situation exacerbated when students' AI prompts were of lower quality. Thus, successful implementation of AI pair programming may require appropriate professional development to prepare instructors to best support their students.

In a study comparing voice-only to embodied AI agents (both in a virtual reality setting), [38] concluded that embodied AI may have more closely approximated a human partner with the corresponding social affordances and therefore increased trust, but only when the AI performed well; otherwise, trust was diminished. [41] notes that students were more productive when they had a plan for solving the programming problem *before* interacting with the AI, as they were better able to prompt the tool and to differentiate correct from incorrect AI output, suggesting that a required planning step might improve AI pair programming outcomes. [30] noted that there were not significant differences between the two different AI tools used.

[42] explored design considerations for embedding AI-assisted coding into a larger learning cycle that included aiding instructors with developing materials as well, finding that their process worked acceptably well. [43] identified important aspects of remote human-human pair programming (e.g., acknowledging the other's contributions, apologizing for mistakes) and then, using a Wizard of Oz prototype, explored which of those aspects were easier or more difficult to reproduce with human-AI pair programming. The most difficult aspects to reproduce were

visualization capabilities, using a breadth-first approach to problem solving, using a depth-first approach, abstracting solutions, using indirect instructions, or asking some types of questions. [42] also observed that traditional pair programming can involve partners with different levels of expertise (i.e., novice and expert, at least relatively), and therefore developed an AI pair programming model where the AI is the expert and the student is the novice, finding that the AI output was appropriate to support student learning (although note that this was in the context of design-based research and was not empirically tested with students). Thus, one of the open questions for AI pair programming is the extent to which the AI should level its output to match the student's programming ability level.

4.2 Recommendations for Supporting AI Pair Programming

Given the novelty of AI pair programming and related research, recommendations for supporting AI pair programming should be considered as preliminary. As a result, our first recommendation is that more research about AI pair programming – particularly at the elementary and secondary levels – should be conducted. Similarly, more research with larger numbers of participants, with a variety of programming languages, and with different AI tools would help expand the understanding of AI pair programming. Second, we note that promising practices for AI pair programming include teacher mediation and embodied AI. Finally, awareness of the negative outcomes found in some studies (e.g., student perceptions of a decreased need to learn to program and reduced engagement, empathy, and social presence) may help identify ways to mitigate these challenges when implementing AI pair programming.

Finally, we note that there is little evidence that students are being prepared for AI pair programming (e.g., with instruction about the affordances and limitations of AI systems); this preparation may lead to improved learning outcomes; regardless, it is appropriate to include it in CS classes. This preparation may also include preparation for metacognitive monitoring since a common concern with student AI use is over-reliance.

4.3 A Researcher Note: Using AI in the SLR Process

In order to support researchers conducting systematic literature reviews, particularly if the use of AI tools is under consideration, we share some observations about our search process for this project.

Figure 2 illustrates the sources (i.e., database or AI tool results) of included articles. Most included papers (7 out of 10) were in the Google Scholar results, while Asta included 3, Elicit 4, and IEEE Xplore 2. Google Scholar identified articles published by IEEE that were *not* found by searching IEEE. No included articles were in the results from the ACM Digital Library, the ASEE Paper Repository, or Science Direct. While the two AI tools did find relevant studies, we note concerns about the transparency [45] and reproducibility [46] of AI-generated output – as well as ethical concerns when AI is used in education research [47] or general ethical concerns related to AI [48] – that may outweigh the case for using AI tools to support literature reviews.

We also note that results that may not match the search string or prompt, offered by both traditional database Google Scholar and by AI tool Elicit, may present challenges in the context of a systematic literature review, where reviewers neither want to exclude relevant materials nor want to evaluate a near-infinite list of irrelevant results. A feature that restricts results to those that specifically match the search criteria would be helpful.

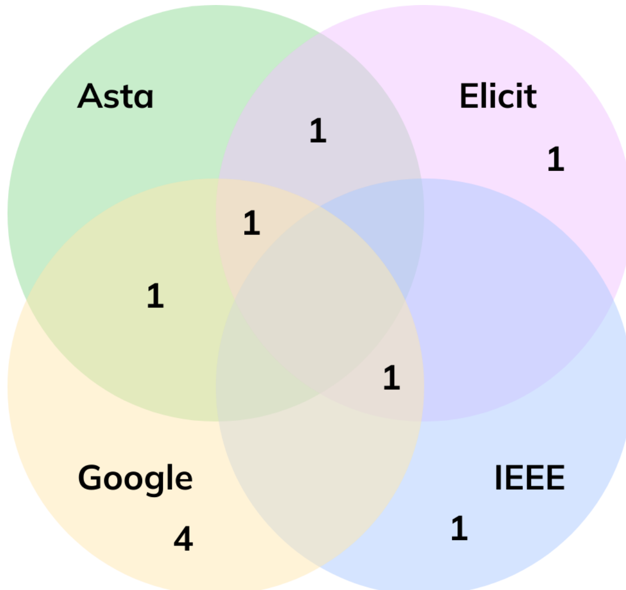

Figure 2: Source(s) of Included Articles; note that no included articles were identified by the ACM Digital Library, ASEE Paper Repository, or Science Direct

For work specifically related to AI pair programming, it would perhaps be helpful to the development of the research base if consistent terminology, especially keywords, were used. Note that [30] uses the keyword "AI-assisted pair programming," while [31] has the phrase "AI-assisted programming" in the title with no keywords related specifically to AI pair programming. Inconsistency in terminology may hamper systematic literature reviews on the topic. More consistency may improve the performance of traditional database searches and thus reduce the incentive to use AI tools, which – as mentioned above – present significant concerns.

Additionally, researchers who are seeking relevant literature but not conducting systematic literature reviews may be interested to know which search options yield the best return on time invested. Thus, we note that the percentage of the total articles found that were ultimately included in the study ranged from 25% (IEEE Xplore) to 0% (ScienceDirect, ACM Digital Library, and ASEE Paper Repository). In the middle of that range were Google Scholar (23%), Elicit (20%), and Asta (10%).

5 Conclusions

We have presented what we believe to be the first systematic literature review specifically focused on the educational use of AI pair programming. We identified a relatively small body of literature that nonetheless showcased a wide variety of methodological approaches and interventions. More research is clearly needed to confirm and to contextualize what has been found so far. We note that the impact of AI pair programming on students appears to be mixed, with some evidence of improvement in learning outcomes existing alongside some negative findings related to affective factors.

We also note that the rapid advances in AI present some challenges to the usual cadence of academic publishing: by the time a study has been published, whatever AI tool was used by students to pair program has likely been replaced by a newer version, raising issues about the applicability of the study's findings. One way to mitigate this problem might be to study whether there are significant performance differences for AI pair programming between older and newer models of AI tools.

6 Acknowledgments

This material is based upon work supported by the National Science Foundation under Award No. 2524429. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Maya Satratzemi, Xinogalos Stelios, and Despina Tsompanoudi. Distributed Pair Programming in Higher Education: A Systematic Literature Review. *Journal of Educational Computing Research*, 61(3):546–577, June 2023. ISSN 0735-6331. doi: 10.1177/07356331221122884. URL https://doi.org/10.1177/07356331221122884. Publisher: SAGE Publications Inc.

[2] Anja Hawlitschek, Sarah Berndt, and Sandra Schulz. Empirical research on pair programming in higher education: a literature review. *Computer Science Education*, 33(3):400–428, July 2023. ISSN 0899-3408. doi: 10.1080/08993408.2022.2039504. URL https://doi.org/10.1080/08993408.2022.2039504. Publisher: Routledge _eprint: https://doi.org/10.1080/08993408.2022.2039504.

[3] Brian Hanks, Sue Fitzgerald, Renée McCauley, Laurie Murphy, and Carol Zander. Pair programming in education: a literature review. *Computer Science Education*, 21(2):135–173, June 2011. ISSN 0899-3408. doi: 10.1080/08993408.2011.579808. URL https://doi.org/10.1080/08993408.2011.579808. Publisher: Routledge _eprint: https://doi.org/10.1080/08993408.2011.579808.

[4] Kuljit Kaur Chahal, Amanpreet Kaur, and Munish Saini. Empirical Studies on Using Pair Programming as a Pedagogical Tool in Higher Education Courses:: A Systematic Literature Review. In *Research and Evidence in Software Engineering*. Auerbach Publications, 2021. Num Pages: 36.

[5] Fan Xu and Ana-Paula Correia. Adopting distributed pair programming as an effective team learning activity: a systematic review. *Journal of Computing in Higher Education*, 36(2):320–349, August 2024. ISSN 1867-1233. doi: 10.1007/s12528-023-09356-3. URL https://doi.org/10.1007/s12528-023-09356-3.

[6] Nor Zalina Ismail and Mohd Rizal Razak. Pair programming for learning programming subject in higher education: A systematic review. *International Journal of Education, Psychology and Counseling (IJEPC)*, 9 (55), September 2024. ISSN 0128-164X. URL https://gaexcellence.com/ijepc/article/view/4284.

[7] Norsaremah Salleh, Emilia Mendes, and John C. Grundy. Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review. *IEEE Transactions on Software Engineering*, 37(04):509–525, July 2011. ISSN 0098-5589. doi: 10.1109/TSE.2010.59. URL https://www.computer.org/csdl/journal/ts/2011/04/tts2011040509/13rRUxC0SQq. Publisher: IEEE Computer Society.

[8] Yeliz Tunga and Saniye Tokel. *The use of pair programming in education: A systematic literature review*. September 2018.

[9] Mark Turner, Rumjit Kaur, and Pearl Brereton. A Lightweight Systematic Literature Review of Studies about the use of Pair Programming to Teach Introductory Programming. January 2008.

[10] Norsaremah Salleh. A Systematic Review of Pair Programming Research-Initial Results. *New Zealand Computer Science Research Student Conference, NZCSRSC 2008 - Proceedings*, January 2008.

[11] Zuraihah Ngadengon, Z Subramaniam, T Yasak, Z Syukri, and Noor Hazim. Computational Thinking and Pair Programming: A Systematic Review. 6:103–119, March 2024. doi: 10.35631/IJMOE.620008.

[12] Nicholas A. Bowman, Lindsay Jarratt, K.C. Culver, and Alberto Maria Segre. How Prior Programming Experience Affects Students' Pair Programming Experiences and Outcomes. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '19, pages 170–175, New York, NY, USA, July 2019. Association for Computing Machinery. ISBN 978-1-4503-6895-7. doi: 10.1145/3304221.3319781. URL `https://dl.acm.org/doi/10.1145/3304221.3319781`.

[13] Caroline Lott, Alexander McAuliffe, and Sandeep Kaur Kuttal. Remote Pair Collaborations of CS Students: Leaving Women Behind? In *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–11, October 2021. doi: 10.1109/VL/HCC51201.2021.9576394. URL `https://ieeexplore.ieee.org/document/9576394`. ISSN: 1943-6106.

[14] Juho Leinonen, Paul Denny, Stephen MacNeil, Sami Sarsa, Seth Bernstein, Joanne Kim, Andrew Tran, and Arto Hellas. 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*, ITiCSE 2023, pages 124–130, New York, NY, USA, June 2023. Association for Computing Machinery. ISBN 979-8-4007-0138-2. doi: 10.1145/3587102.3588785. URL `https://dl.acm.org/doi/10.1145/3587102.3588785`.

[15] Natalie Kiesler, Jacqueline Smith, Juho Leinonen, Armando Fox, Stephen MacNeil, and Petri Ihantola. The Role of Generative AI in Software Student CollaborAItion. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, ITiCSE 2025, pages 72–78, New York, NY, USA, June 2025. Association for Computing Machinery. ISBN 979-8-4007-1567-9. doi: 10.1145/3724363.3729040. URL `https://dl.acm.org/doi/10.1145/3724363.3729040`.

[16] Jiangyue Liu and Siran Li. Toward Artificial Intelligence-Human Paired Programming: A Review of the Educational Applications and Research on Artificial Intelligence Code-Generation Tools. *Journal of Educational Computing Research*, 62(5):1385–1415, September 2024. ISSN 0735-6331. doi: 10.1177/07356331241240460. URL `https://doi.org/10.1177/07356331241240460`. Publisher: SAGE Publications Inc.

[17] Qianou Ma, Tongshuang Wu, and Kenneth Koedinger. Is AI the better programming partner? Human-Human Pair Programming vs. Human-AI pAIr Programming, June 2023. URL `http://arxiv.org/abs/2306.05153`. arXiv:2306.05153 [cs].

[18] Inbal Shani. Survey reveals AI's impact on the developer experience, June 2023. URL `https://github.blog/2023-06-13-survey-reveals-ais-impact-on-the-developer-experience/`.

[19] Joel Becker, Nate Rush, Elizabeth Barnes, and David Rein. Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity, July 2025. URL `http://arxiv.org/abs/2507.09089`. arXiv:2507.09089 [cs].

[20] Shiri Melumad and Jin Ho Yun. Experimental evidence of the effects of large language models versus web search on depth of learning. *PNAS Nexus*, 4(10):pgaf316, October 2025. ISSN 2752-6542. doi: 10.1093/pnasnexus/pgaf316. URL `https://doi.org/10.1093/pnasnexus/pgaf316`.

[21] Yizhou Fan, Luzhen Tang, Huixiao Le, Kejie Shen, Shufang Tan, Yueying Zhao, Yuan Shen, Xinyu Li, and Dragan Gašević. Beware of metacognitive laziness: Effects of generative artificial intelligence on learning motivation, processes, and performance. *British Journal of Educational Technology*, 56(2):489–530, March 2025. ISSN 0007-1013, 1467-8535. doi: 10.1111/bjet.13544. URL `https://bera-journals.onlinelibrary.wiley.com/doi/10.1111/bjet.13544`.

[22] Chunpeng Zhai, Santoso Wibowo, and Lily D. Li. The effects of over-reliance on AI dialogue systems on students' cognitive abilities: a systematic review. *Smart Learning Environments*, 11(1):28, June 2024. ISSN

2196-7091. doi: 10.1186/s40561-024-00316-7. URL
https://doi.org/10.1186/s40561-024-00316-7.

[23] James Prather, Brent N. Reeves, Paul Denny, Brett A. Becker, Juho Leinonen, Andrew Luxton-Reilly, Garrett Powell, James Finnie-Ansley, and Eddie Antonio Santos. “It’s Weird That it Knows What I Want”: Usability and Interactions with Copilot for Novice Programmers. *ACM Trans. Comput.-Hum. Interact.*, 31(1):4:1–4:31, November 2023. ISSN 1073-0516. doi: 10.1145/3617367. URL https://doi.org/10.1145/3617367.

[24] Matthias Hümmer. On the Influence of Artificial Intelligence on Human Problem-Solving: Empirical Insights for Wave 1 from a Multinational Longitudinal Pilot Study, November 2025. URL https://osf.io/preprints/socarxiv/75h3t_v1/.

[25] Arghavan Moradi Dakhel, Vahid Majdinasab, Amin Nikanjam, Foutse Khomh, Michel C. Desmarais, and Zhen Ming (Jack) Jiang. GitHub Copilot AI pair programmer: Asset or Liability? *Journal of Systems and Software*, 203:111734, September 2023. ISSN 0164-1212. doi: 10.1016/j.jss.2023.111734. URL https://www.sciencedirect.com/science/article/pii/S0164121223001292.

[26] Hamsa Bastani, Osbert Bastani, Alp Sungu, Haosen Ge, Özge Kabakcı, and Rei Mariman. Generative AI Can Harm Learning, July 2024. URL https://papers.ssrn.com/abstract=4895486.

[27] Jin Wang and Wenxiang Fan. The effect of ChatGPT on students’ learning performance, learning perception, and higher-order thinking: insights from a meta-analysis. *Humanities and Social Sciences Communications*, 12 (1):621, May 2025. ISSN 2662-9992. doi: 10.1057/s41599-025-04787-y. URL https://www.nature.com/articles/s41599-025-04787-y. Publisher: Palgrave.

[28] Ruiqi Deng, Maoli Jiang, Xinlu Yu, Yuyan Lu, and Shasha Liu. Does ChatGPT enhance student learning? A systematic review and meta-analysis of experimental studies. *Computers & Education*, 227:105224, April 2025. ISSN 0360-1315. doi: 10.1016/j.compedu.2024.105224. URL https://www.sciencedirect.com/science/article/pii/S0360131524002380.

[29] Majeed Kazemitabaar, Justin Chow, Carl Ka To Ma, Barbara J. Ericson, David Weintrop, and Tovi Grossman. 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*, CHI ’23, pages 1–23, New York, NY, USA, April 2023. Association for Computing Machinery. ISBN 978-1-4503-9421-5. doi: 10.1145/3544548.3580919. URL https://doi.org/10.1145/3544548.3580919.

[30] Guangrui Fan, Dandan Liu, Rui Zhang, and Lihu Pan. The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: a comparative study with traditional pair programming and individual approaches. *International Journal of STEM Education*, 12(1): 16, March 2025. ISSN 2196-7822. doi: 10.1186/s40594-025-00537-3. URL https://doi.org/10.1186/s40594-025-00537-3.

[31] Marcel Valový and Alena Buchalcevova. The psychological effects of AI-assisted programming on students and professionals. In *2023 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 385–390, October 2023. doi: 10.1109/ICSME58846.2023.00050. URL https://ieeexplore.ieee.org/document/10336271. ISSN: 2576-3148.

[32] Wenhan Lyu, Yimeng Wang, Yifan Sun, and Yixuan Zhang. Will your next pair programming partner be human? An empirical evaluation of generative AI as a collaborative teammate in a semester-long classroom setting. In *Proceedings of the Twelfth ACM Conference on Learning @ Scale*, L@S ’25, pages 83–94, New York, NY, USA, July 2025. Association for Computing Machinery. ISBN 979-8-4007-1291-3. doi: 10.1145/3698205.3729544. URL https://dl.acm.org/doi/10.1145/3698205.3729544.

[33] Khalid S Khan, Regina Kunz, Jos Kleijnen, and Gerd Antes. Five steps to conducting a systematic review. *Journal of the Royal Society of Medicine*, 96(3):118–121, March 2003. ISSN 0141-0768. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC539417/.

[34] Donald M. Johnson, Will Doss, and Christopher M. Estepp. Agriculture students' use of generative artificial intelligence for microcontroller programming. *Natural Sciences Education*, 53(2):e20155, 2024. ISSN 2168-8281. doi: 10.1002/nse2.20155. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nse2.20155. _eprint: https://acsess.onlinelibrary.wiley.com/doi/pdf/10.1002/nse2.20155.

[35] Johan Petersson. Enhancing programming pair workshops: The case of teacher pre-prompting. *Proceedings of 2024 AIS SIGED International Conference on Information Systems Education and Research*, March 2025. URL https://aisel.aisnet.org/siged2024/15.

[36] Lin Zhang, Qiang Jiang, Weiyan Xiong, and Wei Zhao. Effects of ChatGPT-based human–computer dialogic interaction programming activities on student engagement. *Journal of Educational Computing Research*, 63(4): 988–1023, July 2025. ISSN 0735-6331. doi: 10.1177/07356331251333874. URL https://doi.org/10.1177/07356331251333874. Publisher: SAGE Publications Inc.

[37] Otávio Lube Dos Santos and Davidson Cury. Challenging the confirmation bias: Using ChatGPT as a virtual peer for peer instruction in computer programming education. In *2023 IEEE Frontiers in Education Conference (FIE)*, pages 1–7, October 2023. doi: 10.1109/FIE58773.2023.10343247. URL https://ieeexplore.ieee.org/abstract/document/10343247. ISSN: 2377-634X.

[38] Xiaoran Yang, Yang Zhan, Noboru Matsuda, and Qiao Jin. From tool to partner: Exploring the roles of embodiment on AI agent in pair programming. In *The IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'25)*, September 2025.

[39] Bojan Banić, Mario Konecki, and Mladen Konecki. Pair programming education aided by ChatGPT. In *2023 46th MIPRO ICT and Electronics Convention (MIPRO)*, pages 911–915. IEEE, 2023. URL https://ieeexplore.ieee.org/abstract/document/10159727/.

[40] Mario Simaremare, Chandro Pardede, Irma Tampubolon, Putri Manurung, and Daniel Simangunsong. Pair programming in programming courses in the era of generative AI: students' perspective. In *2024 31st Asia-Pacific Software Engineering Conference (APSEC)*, pages 507–511, December 2024. doi: 10.1109/APSEC65559.2024.00069. URL https://ieeexplore.ieee.org/document/10967301. ISSN: 2640-0715.

[41] Selin Urhan and Selay Arkun Kocadere. Problem-solving through pair-programming: The mediational role of ChatGPT. In *2024 5th International Conference in Electronic Engineering, Information Technology & Education (EEITE)*, pages 1–10, May 2024. doi: 10.1109/EEITE61750.2024.10654434. URL https://ieeexplore.ieee.org/document/10654434.

[42] Zhengyuan Wei, Albert T.L. Lee, Victor C.S. Lee, and Wing-Kwong Chan. Toward AI-facilitated learning cycle in integration course through pair programming with AI agents. In *2024 36th International Conference on Software Engineering Education and Training (CSEE&T)*, pages 1–5, July 2024. doi: 10.1109/CSEET62301.2024.10663037. URL https://ieeexplore.ieee.org/document/10663037. ISSN: 2377-570X.

[43] Sandeep Kaur Kuttal, Jarow Myers, Sam Gurka, David Magar, David Piorkowski, and Rachel Bellamy. Towards designing conversational agents for pair programming: Accounting for creativity strategies and conversational styles. *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–11, August 2020. doi: 10.1109/VL/HCC50065.2020.9127276. URL https://ieeexplore.ieee.org/document/9127276/. Conference Name: 2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) ISBN: 9781728169019 Place: Dunedin, New Zealand Publisher: IEEE.

[44] Yin-Chan Liao, Jiyoung Kim, Anne T. Ottenbreit-Leftwich, Michael Karlin, and Meize Guo. Voices of Elementary Computer Science Teachers: Computer Science Integration Rationales and Practices. *ACM Trans. Comput. Educ.*, 24(4):43:1–43:26, October 2024. doi: 10.1145/3688854. URL https://doi.org/10.1145/3688854.

[45] Francisco Bolaños, Angelo Salatino, Francesco Osborne, and Enrico Motta. Artificial intelligence for literature reviews: opportunities and challenges. *Artificial Intelligence Review*, 57, August 2024. doi: 10.1007/s10462-024-10902-3.

[46] Olivia Guest, Marcela Suarez, Barbara Müller, Edwin van Meerkerk, Arnoud Oude Groote Beverborg, Ronald de Haan, Andrea Reyes Elizondo, Mark Blokpoel, Natalia Scharfenberg, Annelies Kleinherenbrink, Ileana Camerino, Marieke Woensdregt, Dagmar Monett, Jed Brown, Lucy Avraamidou, Juliette Alenda-Demoutiez, Felienne Hermans, and Iris van Rooij. Against the Uncritical Adoption of 'AI' Technologies in Academia, September 2025. URL `https://zenodo.org/records/17065099`.

[47] Lisa Messeri and M. J. Crockett. Artificial intelligence and illusions of understanding in scientific research. *Nature*, 627(8002):49–58, March 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07146-0. URL `https://www.nature.com/articles/s41586-024-07146-0`. Publisher: Nature Publishing Group.

[48] Munmun Ghosh. Artificial intelligence (AI) and ethical concerns: a review and research agenda. *Cogent Business & Management*, 12(1):2551809, December 2025. ISSN 2331-1975. doi: 10.1080/23311975.2025.2551809. URL `https://www.tandfonline.com/doi/full/10.1080/23311975.2025.2551809`.