Understanding Help-Seeking Behavior of Students Using LLMs vs. Web Search for Writing SQL Queries

Harsh Kumar
University of Toronto
Toronto, Canada
harsh@cs.toronto.edu

Mohi Reza
University of Toronto
Toronto, Canada
mohireza@cs.toronto.edu

Jeb Thomas-Mitchell
University of Toronto
Toronto, Canada
jeb.thomasmitchell@mail.utoronto.ca

Ilya Musabirov
University of British Columbia
Vancouver, Canada
ilm@cs.ubc.ca

Lisa Zhang
University of Toronto Mississauga
Mississauga, Canada
lczhang@cs.toronto.edu

Michael Liut
University of Toronto Mississauga
Mississauga, Canada
michael.liut@utoronto.ca

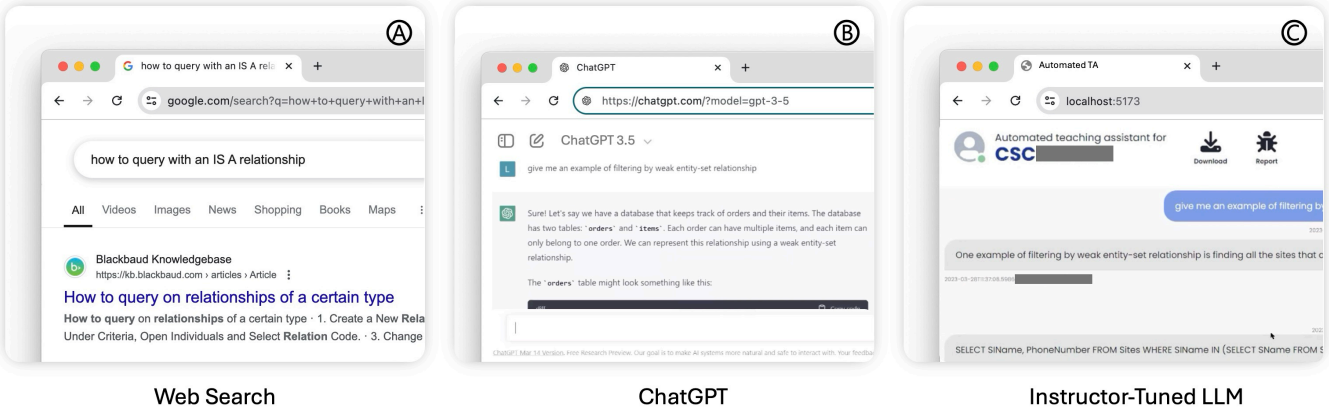


Figure 1: Through a randomized interview study in an SQL course, we compared how students use web search (A) with an off-the-shelf LLM (B) and an instructor-tuned LLM chatbot (C) that has knowledge about the course context and content.

Abstract

Growth in the use of large language models (LLMs) in programming education is altering how students write SQL queries. Traditionally, students relied heavily on web search for coding assistance, but this has shifted with the adoption of LLMs like ChatGPT. However, the comparative process and outcomes of using web search versus LLMs for coding help remain underexplored. To address this, we conducted a randomized interview study in a database classroom to compare web search and LLMs, including a publicly available LLM (ChatGPT) and an instructor-tuned LLM, for writing SQL queries. Our findings indicate that using an instructor-tuned LLM required significantly more interactions than both ChatGPT and web search, but resulted in a similar number of edits to the final SQL query. No significant differences were found in the quality of the final SQL queries between conditions, although the LLM conditions directionally showed higher query quality. Furthermore, students using instructor-tuned LLM reported a lower mental demand. These results have implications for learning and productivity in programming education.

CCS Concepts

• **Applied computing** → **Computer-assisted instruction**; • **Human-centered computing** → **Empirical studies in HCI**; • **Social and professional topics** → **Computing literacy**.

Keywords

Structured Query Language, Large Language Models, Data Systems Education, Human-Computer Interaction, Experiments

ACM Reference Format:
Harsh Kumar, Mohi Reza, Jeb Thomas-Mitchell, Ilya Musabirov, Lisa Zhang, and Michael Liut. 2025. Understanding Help-Seeking Behavior of Students Using LLMs vs. Web Search for Writing SQL Queries. In *Workshop on Data Systems Education: Bridging Education Practice (DataEd '25), June 22–27, 2025, Berlin, Germany*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3735091.3737569

[*] Both authors contributed equally to this research.

DataEd '25, Berlin, Germany

ACM ISBN 979-8-4007-1918-9/25/06
https://doi.org/10.1145/3735091.3737569

1 Introduction

A recent survey [8] of over three thousand programmers revealed that 84% are using AI tools, with ChatGPT being the most popular—74.9% of developers use it on a weekly basis. The most popular use-case (over 80%) is using these tools as a *search engine* for new topics, highlighting the potential for LLM-based chatbots to supplement the programming learning process alongside conventional search methods. For structured tasks such as formulating SQL queries, LLM chatbots offer the unique ability to generate tailored responses coupled with explanations. This contrasts with traditional search methods, where learners must hunt for and adapt code snippets from sites like Stack Overflow.

Despite the growing popularity of LLM chatbots for search-engine-like tasks, the comparative impact of traditional web search versus LLM chatbots in real-world programming classrooms, particularly in supplementing student engagement and learning of languages like SQL, is not well-understood. Empirical insights into how students engage with these tools can inform the design of better learning support tools, especially in flipped classroom settings where self-regulated learning through out-of-class practice plays an important role. Furthermore, exploring whether tuning off-the shelf chatbots like ChatGPT using cost-effective and easy methods, such as adding system prompts, can enhance their usefulness for students is an open question that is of interest to many instructors who are navigating the integration of AI tools into their teaching, while trying to mitigate many of the common concerns that off-the-shelf LLM chatbots pose, like their tendency to generate direct answers, or their lack of awareness of course specific content and context.

To address these issues, we conducted a mixed-methods randomized interview study with 39 students where we compared traditional search (e.g., Google, Bing) with both standard ChatGPT (3.5 model, which was the frontier at the time of the experiment) and an instructor-tuned version of the chatbot with added guard rails (e.g. being told not to give out direct answers) and course context (e.g. a description of the learning goals and content covered by the course). We asked two research questions:

- **RQ1:** How does traditional web search (e.g., Google, Bing) compare to LLM-based chatbots like ChatGPT in terms of student engagement and learning outcomes in programming education, particularly for languages like SQL?
- **RQ2** Can low-cost tuning methods, such as adding system prompts, be employed by instructors to enhance the effectiveness of LLM-based chatbots like ChatGPT for educational purposes, and if so, how does this tuning influence student engagement and learning outcomes?

We found that students interacted with the instructor-tuned LLM more than twice as much compared to both standard ChatGPT ($p = 0.01$) and web search ($p < 0.0001$). Despite this increased engagement, there were no significant differences in the correctness of the final SQL queries across conditions, although the LLM conditions showed higher query quality directionally. These results suggest the potential value of domain experts tuning LLMs using inexpensive methods like system prompts to enhance learner engagement.

The main contributions of this work are:

- Findings from a randomized experiment in a real-world SQL classroom comparing web search with plain ChatGPT and an instructor-tuned LLM, demonstrating how instructor-tuning can significantly impact engagement.
- A useful snapshot of GPT-3.5 usage for SQL education, serving as a benchmark for future studies with newer models.
- Empirical insights into web-search and LLM chatbot usage that can inform the design of learning and productivity tools for programming education using AI.

2 Related Work

The advent of large language models (LLMs) has prompted comparative research with traditional web search methods for information retrieval and problem solving. Recent studies have compared web search and LLM querying for general information-seeking tasks. Wazzan *et al.* found that web search outperformed LLMs in accuracy in a geolocation task, with LLM users struggling to formulate effective queries [29]. Xu observed that LLM users spent less time on their tasks and demonstrated more consistent performance across education levels, but accuracy suffered in fact-checking and complex tasks compared to participants using web search [32]. Spatharioti *et al.* reported that LLM users completed product comparison tasks more quickly by using fewer, more complex queries [26]. While LLM users generally reported higher satisfaction and perceived response quality [26, 32], their accuracy was dependent on the reliability of LLM-provided information and effective prompting [26, 32].

Despite widespread adoption of LLMs in programming tasks [9, 10, 12, 15, 21], fewer comparative studies have been conducted between web search and LLM querying in computer science and computer science education. Research shows that professionals select between web search and querying LLMs through search strategies that utilize self-reflection on their knowledge of the problem and domain [33]. Yen *et al.* found that web search is preferred when professionals are unfamiliar with the domain, or the problem is poorly-defined, because it returns greater diversity of results than LLM queries. Using an LLM is preferred when the user believes the problem is discussed frequently enough that the model has awareness of it, and they possess sufficient knowledge to identify incorrect responses [33]. In contrast, research into students' use of web-based resources showed that students often exhibit shallow, trial-and-error approaches without clear strategy or self-reflection [30]. While some students use web search for general reference [24], other students tend to seek quick answers and exact code matches to specific problems [30]. They tend to exhibit a production bias where they focus on solving the immediate task by quick web searches rather than engaging with foundational concepts in course resources [30]. LLMs provide a potential solution to these strategy and orientation problems. Students view LLMs as providing more specific, easily understandable responses [24] and utilize them for various programming education tasks like generating practice exercises, clarifying error messages, or providing tips on syntax [5].

However, for programming novices, both approaches have potential pitfalls. Web searches can lead to the faulty integration of poorly-understood code, resulting in compounding errors [25].

Students' limited technical vocabulary impedes their ability to define the problem sufficiently for effective keyword-based searches [30]. Students often rely on complex resources like StackOverflow, which they may find difficult to comprehend as novices [30]. With LLMs, there are concerns about overreliance potentially hindering the development of critical thinking and problem-solving skills [5]. Students express beliefs that using LLMs is "closer to cheating" and "doesn't really teach [them] anything" [24]. Additionally, LLM users may become stuck iterating on prompts that produce incorrect results [33].

Despite the challenges, LLMs show promise in enhancing students' understanding of programming concepts, if their reliability can be improved [14]. They offer interactive, beginner-friendly explanations and can provide tailored support for various aspects of programming education [5, 24]. However, there is a clear need for more research directly comparing student use of web search and LLMs in the context of computer science education. This research will enable a more effective integration of these technologies and inform future pedagogical approaches.

3 Methods

We conducted a mixed-methods randomized interview study with 39 students where we compared traditional web search with both standard ChatGPT (3.5 model, which was the frontier at the time of the experiment) and an instructor-tuned version of the chatbot with added guard rails (e.g. being told not to give out direct answers) and course context (e.g. a description of the learning goals and content covered by the course). Students were provided Entity-Relationship diagrams and then asked to solve two SQL-writing problems, one after the other, with one of the randomly selected source of help for each question.

3.1 Context of Deployment

This research study occurred in a 12-week introductory database systems course at a large, publicly funded, research-intensive university in North America, and received ethics board approval (Ethics Protocol #123456[1]) This course targeted third-year undergraduate students in a four-year honors computing program, with 226 students enrolled in the course. The course used a flipped classroom approach, which allowed students to engage with video content and exercises on a custom Learning Management System (LMS) before attending synchronous in-person lecture sessions. Given the flipped approach, students often sought external resources such as web search and LLM-based chatbots to supplement their self-regulated learning, making this an ideal context for deploying the study.

The study was advertised as an optional activity to test the effectiveness of an LLM-based chatbot tutor. 39 students volunteered to participate and were then enrolled in the interview study. The interviews were conducted over Zoom and lasted almost 30 minutes (each round was time-boxed to 15 minutes). During the interview study, each student was randomly assigned to interact with two sources of help, one after the other, to solve two different types of SQL writing problems (randomized in order). 21 were assigned to the Instructor-tuned LLM vs. Web Search round, while the remaining (18) participated in the Instructor-tuned LLM vs. ChatGPT round.

[1] redacted for review.

3.2 Experimental Conditions

This study considered three distinct sources of help in writing code.

3.2.1 Web Search. Students in this round could freely use any web search engine to find resources to help write the SQL query. All students used Google search and then navigated to coding blogs and forums such as GeeksForGeeks, StackOverflow, etc. Figure 1A shows an example query issued by one of the students in the study.

3.2.2 ChatGPT. In this round, students were directly given access to ChatGPT (3.5). No additional tweaks were made to the model. This acted as a proxy for publicly available LLMs being used by students to solve assignment problems on their own. Figure 1B shows a student's interaction with ChatGPT during the study.

3.2.3 Instructor-tuned LLM. Students in this round were given access to a chatbot using GPT-3.5 (the frontier model at the time of the study). The model was configured with a system prompt to be particularly helpful for writing the SQL queries and provided the context of the questions used in the study. This acted as a proxy for instructor-provided LLM chatbots used in classrooms (e.g., KhanMigo, CS50 bot [19], etc.) Figure 1C shows the chatbot interface used in the study.

3.3 Outcome Measures

Each interview study session was recorded on video, and the following outcome measures were manually extracted from the videos.

3.3.1 Number of Interactions with the Source of Help. We measure this by calculating the number of queries sent to the assigned source of help. Each query constitutes an interaction.

3.3.2 Number of Edits Made to the Final SQL Query. This was measured by calculating the number of changes made to the final SQL query submitted by the student, during each round of the study.

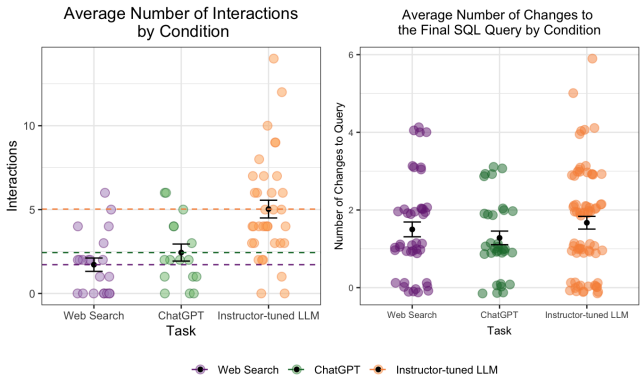
3.3.3 Quality of the SQL Query. We utilized a grading rubric specified by the course instructor for the two types of questions. Based on this rubric, one of the authors assigned a score (0-100) to each of the final SQL queries of the students.

3.3.4 Self-reported Mental Demand. We used the *Mental Demand* subscale from the NASA-TLX questionnaire [22] to compare the mental demands of completing the SQL-writing task with their assigned source of help. On a scale of 1 (very low) to 10 (very high), students were asked to rate how mentally demanding the task was.

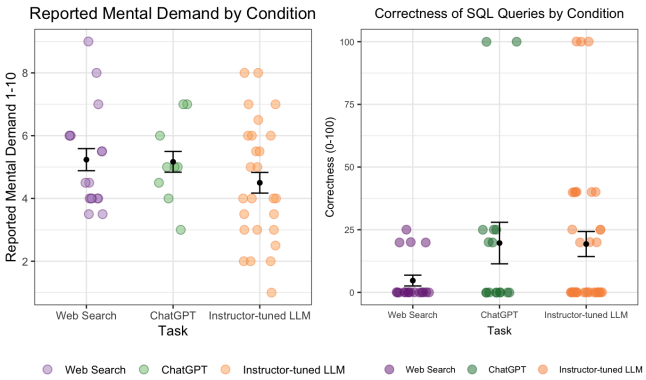
4 Results

4.1 Effect on Number of Interactions

Figure 2a left-facet shows the average number of interactions with the assigned source of help by condition. We performed a Kruskal-Wallis H test to compare the number of interactions across the three conditions and the results indicated a statistically significant difference between the groups ($\chi^2(2) = 20.5$, $df = 2$, $p < 0.0001$).



(a) Number of Interactions and SQL Query Changes.



(b) Mental Demand and Correctness of SQL Queries.

Figure 2: Comparative analyses between conditions. (a) shows the average number of interactions (left panel, higher for Instructor-tuned LLM) and average number of SQL query changes (right panel, no significant differences). (b) shows students' average self-reported mental demand (left panel, no significant differences but directionally lower for Instructor-tuned LLM) and SQL query correctness (right panel, higher with either LLM compared to web search). Error bars represent ±1 SEM.

Post-hoc pairwise comparisons were conducted using Dunn's test with Bonferroni correction. Students interacted with the Instructor-tuned LLM more than twice compared to ChatGPT ($p = 0.01$) and Web Search ($p < 0.0001$).

4.2 Effect on Number of SQL Query Edits/Changes

Figure 2a right-facet shows the average number of changes made to the final SQL query provided by the student. The Kruskal-Wallis H test for changes between conditions was not significant ($\chi^2(2) = 1.55$, $df = 2$, $p = 0.46$). In all conditions, students needed just under 2 changes to arrive at their final query in the assigned time. However, there were students in the Instructor-tuned LLM condition who made over 5 changes in their final SQL query (Figure 2a).

4.3 Effect on the Quality of the Final SQL Query

Figure 2b right-facet shows the average correctness of the final SQL queries by condition. The Kruskal-Wallis H test for the correctness between conditions was not significant ($\chi^2(2) = 3.85$, $df = 2$, $p = 0.14$). However, correctness was directionally higher for both LLM conditions than for the web search condition.

4.4 Self-Reported Mental Demand

The Kruskal-Wallis H test for mental demands was not significant ($\chi^2(2) = 2.05$, $df = 2$, $p = 0.36$). However, there is suggestive evidence that students in the Instructor-tuned LLM conditions reported that the task was less mentally demanding compared to the other conditions (see Figure 2b left-facet). When comparing the instructor-tuned LLM with ChatGPT, one of the students commented *"[instructor-tuned LLM] seem more accurate or specific than chatGPT."* This may indicate lesser mental demand on the student.

5 Discussion

Key Findings. We found that students needed to interact more with the Instructor-tuned LLM compared to ChatGPT and Web Search. This increased interaction could be attributed to the system prompt for the Instructor-tuned LLM, which included sentences such as *"The instructor does not provide the exact answer to the given problem..."*, as well as other guardrails to prevent cheating with LLMs [6, 14]. One might hypothesize that this would lead to lower grades for students using the Instructor-tuned LLM versus ChatGPT and Web Search, which can readily provide direct answers. However, our results did not show significant differences in the quality of the final SQL queries between conditions. This is promising, as greater engagement could potentially lead to longer-term learning [3, 16]. Students expressed interest in using instructor-tuned LLM over ChatGPT and said *"Would rather use [instructor-tuned LLM] over ChatGPT given the prior knowledge of the tables and helps with practical examples of how to join two tables."* This supports the idea that scaffolded learning, in which students are guided but not given direct answers, can be as effective as direct instruction [13, 20].

Interestingly, the higher number of interactions with the Instructor-tuned LLM did not result in higher reported mental demand. In fact, students who used Instructor-tuned LLM reported levels of mental demand that were equal to or lower than those using ChatGPT and Web Search. Lowering the cognitive load can make programming more approachable, potentially reducing dropout rates in CS and encouraging more students to pursue and persist in the field [23, 28]. Additionally, there were no differences in the number of changes made to the final SQL query between the different conditions. This suggests that while the Instructor-tuned LLM may require more interaction, it does not necessarily increase the mental burden on students and maintains the same level of code refinement as other methods [4, 11]. In summary, our findings highlight the potential of using LLMs as facilitators of learning rather than just sources of

information and contribute to the growing literature of designing pedagogically informed LLM-tutors [14, 18].

Broader Implications. Studies such as this show the value of low-cost instructor tuning through system prompting for increasing student engagement. Instructor-tuned chatbots can serve as "levers" for instructors wishing to amplify student engagement in their courses outside of the classroom by offering personalized support to students through chatbots that have been "tuned" with course-specific context and content. The provision of these chatbots may also reduce over-dependency on general-purpose chat agents like ChatGPT [2, 7, 17]. At the same time, the effectiveness of these levers will depend on both the accuracy of the models themselves (something we can expect to improve with future frontier models) and the ability of students to ask the right questions (through prompts, which is harder than it may appear to users [16, 34]). Without the latter, increased engagement may not translate into increased learning, as we saw with the ratings of the quality of the final query in our study. Therefore, providing better support for students by asking the *right* questions is a useful direction for future work. The lower mental demand reported by participants for the instructor-tuned condition also hints at the potential value of instructor-tuning for helping users manage the metacognitive demands of using AI, as found in recent studies [27, 31].

Limitations & Future Work. The small sample size presents a primary threat to validity. Although measures such as randomizing questions and condition orders were implemented to mitigate biases, the limited sample size may still impact the generalizability and power of the findings. The participants in our study were upper-year CS students from a research-intensive university. This further raises questions about the validity of the general population's findings related to SQL query writing. Additionally, the study did not measure long-term learning outcomes, which limits understanding of how the different methods influence sustained learning and retention over time. Moreover, the presence of the interviewer during the programming task may have affected the students' help-seeking behavior [1].

Future work may involve larger-scale, between-subjects, randomized controlled experiments to enhance the generalizability and robustness of the findings, such as through a multi-institutional longitudinal study. Furthermore, investigating the long-term learning outcomes associated with using LLMs versus web search for coding assistance will provide deeper insights into their impacts on sustained learning and retention. Beyond prompting, instructor-tuned LLM can be made even more useful for learning by fine-tuning the language models with pedagogically rich data [14]. Exploring different types of programming tasks and expanding the study to diverse educational settings could further elucidate LLMs' broader applicability and effectiveness in data systems education. Future work should investigate how much *learning* happens when using LLMs for programming, compared to relying on search engines. One might hypothesize that getting direct solutions from LLMs may hamper learning compared to getting clues from search. In this case, instructor-provided LLMs can hold key in balancing the tradeoffs between the benefits of using LLMs with the amount of learning on the students' part.

6 Conclusion

We conducted a randomized interview study to compare students' use of LLMs (out-of-the-box and instructor-tuned) with conventional web search (status quo) for writing SQL queries. Our findings suggest that an instructor-tuned LLM might lead to higher engagement, on average, compared to other sources of help while maintaining the quality of downstream performance on the given task. Preliminary evidence also points to a reduction in students' self-reported mental demand for writing SQL queries while utilizing the instructor-tuned LLM. These findings highlight the potential of designing LLM-based instructional resources with the participation of teachers and have implications for the field of productivity and the future of work, in addition to education technology.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grant #RGPIN-2024-04348, and the Learning & Education Advancement Fund from the Office of the Vice-Provost, Innovations in Undergraduate Education, University of Toronto. Additionally, we would like to thank Marko Choi and Hammad Sheikh for supporting this project.

References

[1] John G Adair. 1984. The Hawthorne effect: a reconsideration of the methodological artifact. *Journal of applied psychology* 69, 2 (1984), 334.
[2] Ibrahim Adeshola and Adeola Praise Adepoju. 2023. The opportunities and challenges of ChatGPT in education. *Interactive Learning Environments* (2023), 1–14.
[3] Hamsa Bastani, Osbert Bastani, Alp Sungu, Haosen Ge, Özge Kabakcı, and Rei Mariman. 2024. Generative AI Can Harm Learning. *Available at SSRN 4895486* (2024).
[4] Jhon Alexander Bueno-Vesga, Xinhao Xu, and Hao He. 2021. The effects of cognitive load on engagement in a virtual reality learning environment. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 645–652.
[5] Doga Cambaz and Xiaoling Zhang. 2024. Use of AI-driven Code Generation Models in Teaching and Learning Programming: a Systematic Literature Review. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 172–178. doi:10.1145/3626252.3630958
[6] Eason Chen, Ray Huang, Han-Shin Chen, Yuen-Hsien Tseng, and Liang-Yi Li. 2023. GPTutor: a ChatGPT-powered programming tool for code explanation. In *International Conference on Artificial Intelligence in Education*. Springer, 321–327.
[7] Debby RE Cotton, Peter A Cotton, and J Reuben Shipway. 2024. Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. *Innovations in education and teaching international* 61, 2 (2024), 228–239.
[8] Daniel Daines-Hutt. 2023. The state of AI tools and coding: 2023 edition. https://zerotomastery.io/blog/the-state-of-ai-tools-and-coding-2023-edition
[9] Paul Denny, James Prather, Brett A Becker, James Finnie-Ansley, Arto Hellas, Juho Leinonen, Andrew Luxton-Reilly, Brent N Reeves, Eddie Antonio Santos, and Sami Sarsa. 2024. Computing education in the era of generative AI. *Commun. ACM* 67, 2 (2024), 56–67.
[10] Tao Dong and Luke Church. [n. d.]. Back to the future: What do historical perspectives on programming environments tell us about LLMs? ([n. d.]).
[11] John Edwards, Kaden Hart, and Christopher Warren. 2022. A practical model of student engagement while programming. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1*. 558–564.
[12] Philip J Guo. 2023. Six Opportunities for scientists and engineers to learn programming using AI Tools such as ChatGPT. *Computing in Science & Engineering* 25, 3 (2023), 73–78.
[13] Cindy E Hmelo-Silver, Ravit Golan Duncan, and Clark A Chinn. 2007. Scaffolding and achievement in problem-based and inquiry learning: a response to Kirschner, Sweller, and. *Educational psychologist* 42, 2 (2007), 99–107.
[14] Irina Jurenka, Markus Kunesch, Kevin R McKee, Daniel Gillick, Shaojian Zhu, Sara Wiltberger, Shubham Milind Phal, Katherine Hermann, Daniel Kasenberg, Avishkar Bhoopchand, et al. 2024. Towards responsible development of generative AI for education: An evaluation-driven approach. *arXiv preprint arXiv:2407.12687* (2024).

[15] Majeed Kazemitabaar, Runlong Ye, Xiaoning Wang, Austin Zachary Henley, Paul Denny, Michelle Craig, and Tovi Grossman. 2024. Codeaid: Evaluating a classroom deployment of an llm-based programming assistant that balances student and educator needs. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–20.

[16] Harsh Kumar, Ilya Musabirov, Mohi Reza, Jiakai Shi, Anastasia Kuzminykh, Joseph Jay Williams, and Michael Liut. 2023. Impact of guidance and interaction strategies for LLM use on Learner Performance and perception. *arXiv preprint arXiv:2310.13712* (2023).

[17] Harsh Kumar, David M Rothschild, Daniel G Goldstein, and Jake M Hofman. 2023. Math Education with Large Language Models: Peril or Promise? *Available at SSRN 4641653* (2023).

[18] Harsh Kumar, Ruiwei Xiao, Benjamin Lawson, Ilya Musabirov, Jiakai Shi, Xinyuan Wang, Huayin Luo, Joseph Jay Williams, Anna N. Rafferty, John Stamper, and Michael Liut. 2024. Supporting Self-Reflection at Scale with Large Language Models: Insights from Randomized Field Experiments in Classrooms. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale* (Atlanta, GA, USA) *(L@S '24)*. Association for Computing Machinery, New York, NY, USA, 86–97. doi:10.1145/3657604.3662042

[19] Rongxin Liu, Carter Zenke, Charlie Liu, Andrew Holmes, Patrick Thornton, and David J Malan. 2024. 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*. 750–756.

[20] Janet Maybin, Neil Mercer, and Barry Stierer. 1992. Scaffolding learning in the classroom. *Thinking voices: The work of the national oracy project* 186 (1992), 195.

[21] James Prather, Paul Denny, Juho Leinonen, Brett A Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, et al. 2023. 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*. 108–159.

[22] Susana Rubio, Eva Díaz, Jesús Martín, and José M Puente. 2004. Evaluation of subjective mental workload: A comparison of SWAT, NASA-TLX, and workload profile methods. *Applied psychology* 53, 1 (2004), 61–86.

[23] Greg Scragg and Jesse Smith. 1998. A study of barriers to women in undergraduate computer science.. In *Proceedings of the twenty-ninth SIGCSE technical symposium on Computer Science Education*. 82–86.

[24] James Skripchuk, John Bacher, Yang Shi, Keith Tran, and Thomas Price. 2024. Novices' Perceptions of Web-Search and AI for Programming. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 1818–1819. doi:10.1145/3626253.3635545

[25] James Skripchuk, Neil Bennett, Jeffrey Zhang, Eric Li, and Thomas Price. 2023. Analysis of Novices' Web-Based Help-Seeking Behavior While Programming. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 945–951. doi:10.1145/3545945.3569852

[26] Sofia Elena Spatharioti, David M. Rothschild, Daniel G. Goldstein, and Jake M. Hofman. 2023. Comparing Traditional and LLM-based Search for Consumer Choice: A Randomized Experiment. arXiv:2307.03744

[27] Lev Tankelevitch, Viktor Kewenig, Auste Simkute, Ava Elizabeth Scott, Advait Sarkar, Abigail Sellen, and Sean Rintel. 2024. The Metacognitive Demands and Opportunities of Generative AI. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 680, 24 pages. doi:10.1145/3613904.3642902

[28] Jennifer Wang and Sepehr Hejazi Moghadam. 2017. Diversity barriers in K-12 computer science education: Structural and social. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education*. 615–620.

[29] Albatool Wazzan, Stephen MacNeil, and Richard Souvenir. 2024. Comparing Traditional and LLM-based Search for Image Geolocation. In *Proceedings of the 2024 Conference on Human Information Interaction and Retrieval* (Sheffield, United Kingdom) *(CHIIR '24)*. Association for Computing Machinery, New York, NY, USA, 291–302. doi:10.1145/3627508.3638305

[30] David Wong-Aitken, Diana Cukierman, and Parmit K. Chilana. 2022. "It Depends on Whether or Not I'm Lucky" How Students in an Introductory Programming Course Discover, Select, and Assess the Utility of Web-Based Resources. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1* (Dublin, Ireland) *(ITiCSE '22)*. Association for Computing Machinery, New York, NY, USA, 512–518. doi:10.1145/3502718.3524751

[31] Yi Wu. 2023. Integrating generative AI in education: how ChatGPT brings challenges for future learning and teaching. *Journal of Advanced Research in Education* 2, 4 (2023), 6–10.

[32] Ruiyun (Rayna) Xu, Yue (Katherine) Feng, and Hailiang Chen. 2023. ChatGPT vs. Google: A Comparative Study of Search Performance and User Experience. doi:10.2139/ssrn.4498671

[33] Ryan Yen, Nicole Sultanum, and Jian Zhao. 2024. To Search or To Gen? Exploring the Synergy between Generative AI and Web Search in Programming. In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems (CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 327, 8 pages. doi:10.1145/3613905.3650867

[34] J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny Can't Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 437, 21 pages. doi:10.1145/3544548.3581388