



Delft University of Technology

Document Version

Final published version

Licence

CC BY

Citation (APA)

Valle Torre, M., Specht, M., & Oertel, C. (2026). LLM Chatbots in High School Programming: Exploring Behaviors and Interventions. In *SAC 2026 - 41st Annual ACM Symposium on Applied Computing* (pp. 79-86). (Proceedings of the ACM Symposium on Applied Computing). Association for Computing Machinery (ACM).
<https://doi.org/10.1145/3748522.3779870>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

In case the licence states "Dutch Copyright Act (Article 25fa)", this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

LLM Chatbots in High School Programming: Exploring Behaviors and Interventions

Manuel Valle Torre
Delft University of Technology
Delft, The Netherlands
m.valletorre@tudelft.nl

Marcus Specht
Delft University of Technology
Delft, The Netherlands
m.m.specht@tudelft.nl

Catharine Oertel
Delft University of Technology
Delft, The Netherlands
c.r.m.m.oertel@tudelft.nl

Abstract

This study uses a Design-Based Research (DBR) cycle to refine the integration of Large Language Models (LLMs) in high school programming education. The initial problem was identified in an Intervention Group where, in an unguided setting, a higher proportion of executive, solution-seeking queries correlated strongly and negatively with exam performance. A contemporaneous Comparison Group demonstrated that without guidance, these unproductive help-seeking patterns do not self-correct, with engagement fluctuating and eventually declining. This insight prompted a mid-course pedagogical intervention in the first group, designed to teach instrumental help-seeking. The subsequent evaluation confirmed the intervention's success, revealing a decrease in executive queries, as well as a shift toward more productive learning workflows. However, this behavioral change did not translate into a statistically significant improvement in exam grades, suggesting that altering tool-use strategies alone may be insufficient to overcome foundational knowledge gaps. The DBR process thus yields a more nuanced principle: the educational value of an LLM depends on a pedagogy that scaffolds help-seeking, but this is only one part of the complex process of learning.

CCS Concepts

• **Applied computing** → **Computer-assisted instruction; Interactive learning environments.**

Keywords

Programming Education, Help-seeking, Large Language Models, K12

ACM Reference Format:

Manuel Valle Torre, Marcus Specht, and Catharine Oertel. 2026. LLM Chatbots in High School Programming: Exploring Behaviors and Interventions. In *The 41st ACM/SIGAPP Symposium on Applied Computing (SAC '26)*, March 23–27, 2026, Thessaloniki, Greece. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3748522.3779870>

1 Introduction

As technology becomes increasingly integral to education and society, the need for widespread computational skills and programming grows [3]. Learning to program, however, presents unique challenges: Students frequently struggle with abstract concepts such

as loops and conditionals, find error messages cryptic, and may default to memorizing syntax rather than understanding programming logic [4]. These difficulties often lead to frustration, reduced motivation, and high dropout rates, highlighting a critical need for timely, individual support [32, 22].

Recent advancements in Large Language Models (LLMs) present promising solutions for overcoming these educational barriers [7]. By providing personalized, real-time assistance, LLMs have the potential to help students better manage cognitive load, clarify confusing syntax, and focus more deeply on programming concepts [21]. At the same time, the integration of LLMs in educational settings carries risks. Over-reliance on AI assistance may weaken students' independent problem-solving skills, encourage superficial learning strategies, and even hinder the development of key cognitive skills like critical thinking when compared to non-AI approaches [35, 6, 9]. Furthermore, potential inaccuracies or misleading responses from LLMs raise concerns about their reliability as educational assistants [7].

While the importance of guiding student use of GenAI is recognized and there are opportunities to foster metacognitive processes [36], there is a lack of research on structured interventions designed to promote effective use of LLMs. Some prior work has explored data-driven guidance systems for help features in tutoring systems [17], but applying these ideas to the context of GenAI requires further investigation. Moreover, most studies focus on university-level learners, leaving high school students less explored [19]. In short, more empirical research is needed to understand the effects and underlying mechanisms of LLM use in education [30].

To address these gaps, this study employs a Design-Based Research (DBR) cycle to investigate and refine the integration of LLMs in a high school programming course. The study is centered on an Intervention Group (IG), where we first identified a core problem: in an unguided setting, a higher frequency of LLM use—particularly for executive, solution-seeking queries—negatively correlated with midterm exam performance [2]. This critical insight informed the design of a targeted pedagogical intervention, which was then implemented and evaluated in the second phase of the course. Data from a contemporaneous Comparison Group (CG) provides additional context for our findings.

The following research questions guide this study:

- **RQ1 - Baseline:** How does the frequency of student interactions with an LLM correlate with their programming performance?
- **RQ2 - Baseline:** How do instrumental vs. executive query types impact learning outcomes in an unguided setting?
- **RQ3 - Intervention:** How does a targeted intervention influence students' help-seeking behaviors and learning outcomes?



This work is licensed under a Creative Commons Attribution 4.0 International License. *SAC '26, Thessaloniki, Greece*

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2294-3/2026/03

<https://doi.org/10.1145/3748522.3779870>

The following sections build the case for our questions by reviewing prior work, detailing our research design, presenting the results of our analysis, and discussing the implications for supporting high school programming learners.

2 Related Work

Effective pedagogical support extends beyond providing direct solutions; it also involves guiding students toward appropriate help-seeking strategies. Help-seeking theory offers a valuable framework for understanding how students request and utilize assistance, which is highly relevant when introducing AI helpers [11]. Help-seeking has been framed as a positive, strategic part of learning, rather than a sign of failure, and distinguished between executive and instrumental help-seeking [18].

Executive help-seeking is considered non-adaptive; the student aims to minimize effort by having someone else (or an AI) complete the task, for example, by asking, “Can you just give me the answer to this problem?” In contrast, instrumental help-seeking is an adaptive, learning-focused strategy where the student requests just enough help (e.g., a hint or explanation) to overcome an impasse and move forward independently [14]. Instrumental help-seeking is strongly associated with better retention and skill transfer [23]. Ultimately, help-seeking behavior is often considered a key aspect of the broader construct of self-regulated learning (SRL) [11].

This active role is vital for learning, as even well-crafted instructional explanations can be ineffective if the learner does not engage with them to develop their own understanding [29]. By asking instrumental questions, students are prepared to engage in cognitive activities, such as generating inferences or revising their mental models. This process helps them integrate new information with prior knowledge, thereby constructing a more robust understanding of the material [15].

LLM-based tutors make this process even more relevant. Whereas prior research on ITS often studied students requesting incremental hints [1], LLMs typically provide complete explanations, akin to bottom-out hints [20]. This fundamental shift from minimal guidance to comprehensive answers makes it essential to reevaluate whether established help-seeking principles remain, and if students can be guided towards more productive learning behaviors [19].

2.1 Programming Education and Large Language Models

Novice programmers must simultaneously learn new syntax and logic while also interpreting error messages [33, 34]. Studies comparing novice and expert coders highlight these differences, showing novices make more errors and struggle to achieve program correctness [5]. Effective help-seeking behaviors are crucial for overcoming these obstacles, yet novices often lack the metacognitive skills or confidence to ask for appropriate assistance [23, 10, 17]. These challenges underscore the need for timely, individualized, and scalable educational support.

LLMs offer scalable, personalized support in programming education, potentially replicating the benefits of human tutors by explaining code and reducing stress. Recent advancements of LLMs offer potential solutions for providing this in programming education [7, 21]. LLMs can generate and explain code, answer questions immediately, potentially reduce stress, and improve task efficiency [25]. Furthermore, some students may feel more comfortable asking basic or repetitive questions to an LLM compared to a human instructor [21, 8].

However, the integration of LLMs like ChatGPT also presents significant challenges and risks [7]. Firstly, these tools can complicate the programming learning experience, giving confusing or advanced information [26]. If students fail to see the usefulness of

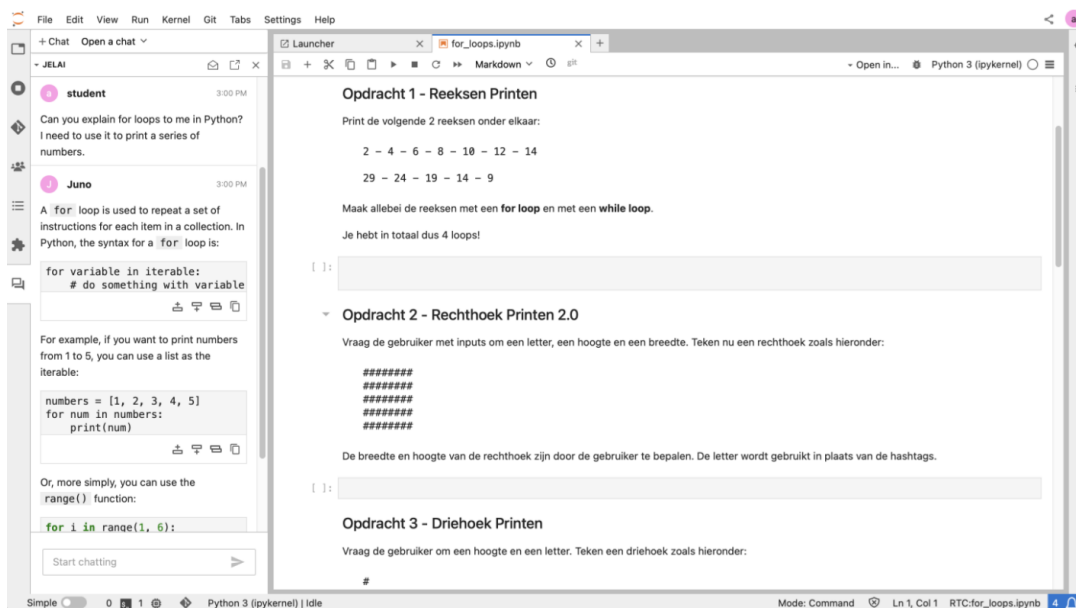


Figure 1: JELAI interface with a sample task in the Jupyter Notebook on the right and a task-specific LLM-based tutor on the left

these systems, they may stop using them before getting any benefits from them [24]. Standard LLMs are not designed as pedagogical tools; they typically prioritize task completion over fostering deeper conceptual understanding [35, 6]. This can lead to superficial engagement and potentially hinder the development of independent problem-solving and debugging skills [25].

Recognizing these complexities, researchers have developed specialized LLM-based systems for programming education, aiming to provide help that supports learning. Some prominent examples are: **CodeTutor**: A chatbot using system prompts for educational responses; its interface is similar to that of ChatGPT and allows students to rate feedback at both the message and conversation levels [16]. **CodeAid**: An AI tutor providing targeted support, guiding the student via specific feature selection while avoiding direct solutions [12]. And **CodeHelp**: A web interface using multiple prompts to ensure an appropriate response from the LLM, which also avoids giving away solutions [25].

While these systems represent important advances, the studies evaluating them often rely on post-task surveys or analysis of conversations in a separate chat window, rather than logging interactions that occur seamlessly within the coding environment itself. The system used in this study, JELAI, is designed to address this by integrating the LLM chatbot directly into the notebook environment, similar to modern development interfaces (like GitHub Copilot or Cursor), potentially making interaction more intuitive.

3 Methodology

3.1 Research Design and Procedure

Integrating LLMs into authentic classroom settings presents a complex, ill-defined problem for which Design-Based Research (DBR) is an ideal methodological framework. DBR enables us to move beyond efficacy questions ("does it work?") to develop a deeper, theoretically grounded understanding of how, why, and under what conditions an intervention functions in a real classroom. Our study follows a classic DBR cycle: we first used help-seeking theory to analyze an existing implementation and identify a core pedagogical problem, then designed and implemented a targeted intervention to address it, with the ultimate goal of generating shareable design principles. The aim of this DBR cycle, therefore, is not statistical generalization to a wider population, but rather the development of a local theory and shareable design principles grounded in the complex, authentic context of this specific classroom.

The design centers on a pre-post analysis of an Intervention Group (IG, $N = 18$) and a non-equivalent Comparison Group (CG, $N = 19$) to contextualize the impact of the intervention. We maintained authentic class sizes to preserve ecological validity and avoid confounding variables (e.g., teacher effects) associated with merging disparate cohorts.

Participants in both groups were recruited from elective Informatica classes at two different Dutch high schools, where they were introduced to foundational programming concepts (e.g., variables, conditionals, loops). The non-equivalent Comparison Group was enlisted to provide context on how unguided help-seeking behaviors evolve over a similar timeframe, rather than for direct statistical comparison. In both settings, students used the JELAI environment

during class assignments under their teacher's supervision. All participants and their parents provided informed consent.

The IG's course was divided into two six-week phases. Participants were asked to complete two short surveys rating their previous experience with programming and with LLM chatbots. In the pre-intervention phase, the unguided use of JELAI by students was logged, and a midterm exam was administered to identify baseline help-seeking behaviors and their correlation with performance. Based on this initial analysis, we implemented a two-part pedagogical intervention. This included a class-wide discussion on the distinction between instrumental and executive help-seeking, followed by brief, one-on-one sessions where students reflected on their own query patterns with a researcher [15]. In the post-intervention phase, students continued the course with JELAI access, and the intervention's effects were evaluated via logs and a final exam.

The CG used JELAI without any specific guidance for their entire course. The data collected included all interactions with the system and their final exam scores; no midterm exams or interventions were administered in this group.

All exams for both groups tested the concepts taught during the lessons, using a mix of syntax and coding tasks in JELAI, but with access to Juno deactivated.

3.2 JELAI Environment and Data Classification

The primary learning environment for both groups was JELAI [28], an open-source platform that integrates an LLM-based chatbot, Juno, directly into a Jupyter Notebook interface (see Figure 1). Juno was powered by a Llama3.1:70b¹ open-weights model, guided by a system prompt² to act as a Socratic tutor that avoids providing direct solutions. JELAI logged all student interactions—including chat messages, code cell executions, and errors—providing a fine-grained dataset of the learning process³. These logs, along with student exam scores, served as the primary data for this study.

To analyze student queries, we employed a multi-step classification process. Drawing from help-seeking theory [18, 14], we first manually coded the IG's pre-intervention queries into three high-level categories: Instrumental (seeking understanding), Executive (seeking solutions), and Other. This manually coded data was used to train and validate a few-shot LLM classifier using DSPy⁴, which achieved an 84% accuracy and was subsequently used to classify all queries from the CG. For a more detailed analysis within the IG, queries were also manually coded into finer-grained sub-categories [25, 31] (see Figure 2).

3.3 Data Analysis

For RQ1, we used Spearman rank correlation (ρ) due to the small sample sizes of our groups and potential non-normality of the data. Similarly, for RQ2, we used Spearman rank correlation (ρ) to assess the relationship between instrumental/executive queries and the grades, as well as a detailed analysis per query type.

For RQ3, investigating the intervention's influence, we employed non-parametric tests: the Wilcoxon signed-rank test was used to

¹<https://ollama.com/library/llama3.1>

²https://github.com/mvallet91/JELAI/blob/celdelta/history_app.py#L88

³The dataset is available at <https://data.4tu.nl/>

⁴<https://ds.py.ai/>

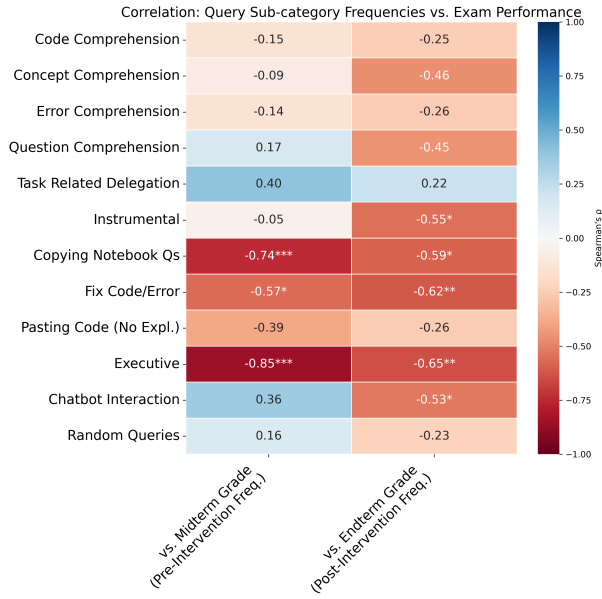


Figure 2: Correlations between student interaction types and grades, left column is data until the midterm correlated with the midterm grades, right column is the data after the midterm, correlated with the final exam grades. (* $p < 0.05$, ** $p < 0.01$, * $p < 0.001$)**

compare paired data (e.g., proportion of query types before and after the intervention for the same students, and midterm vs. final exam scores). All statistical tests were conducted with a significance level of $\alpha = 0.05$.

Based on coding and dialog logs, we describe the learning activities of several participants to contextualize our findings and illuminate specific behaviors. We also conduct sequence analysis on the entire group to further explore the behavioral changes. We describe these insights in the Discussion (in Section 5).

4 Results

This section presents the findings organized by our research questions, first establishing the baseline behaviors observed in both groups and then evaluating the impact of the intervention. Students in the Intervention Group (IG, $N=18$) submitted 480 queries to the LLM, with a mean of 25.4 queries per student. The Comparison Group (CG, $N=19$) submitted 403 queries, with a similar mean of 21.21 queries per user.

Based on pre-course survey for IG, the students were largely novice programmers, reporting a low average prior programming experience ($M = 1.89$, $SD = 1.02$) on a 5-point scale. Their self-reported prior usage of LLM chatbots was moderate, corresponding to an average use of 1-2 times per week ($M = 3.06$, $SD = 1.11$) on a 5-point scale.

RQ1: Correlation Between Interaction Frequency and Performance In the IG's pre-intervention phase, we found a significant, negative correlation between the total number of LLM queries and midterm exam scores ($\rho = -0.502$, $p = 0.034$), indicating that

higher interaction frequency was associated with lower performance. This is consistent with prior ITS research suggesting that frequency often signals a struggling student [2]. Further analysis suggested this was a selection effect; after controlling for students' self-reported prior programming experience, which showed a negative trend with interaction frequency ($\rho = -0.442$, $p = 0.067$), the direct relationship between frequency and grades became non-significant ($p = 0.084$). In the CG, no significant correlation was found between total query frequency and final exam grades ($\rho = -0.059$, $p = 0.811$)

RQ2: Impact of Instrumental vs. Executive Queries Analysis of query types in the IG's pre-intervention phase revealed the core problem for our DBR cycle. The most frequent query type was Concept Comprehension (24.7%), followed by Copying Notebook Questions (19.2%), a clear executive behavior. This was reflected in the correlations: there was a clear and significant negative correlation between the proportion of executive queries and midterm grades ($\rho = -0.851$, $p < 0.001$), while instrumental queries showed no significant correlation ($\rho = -0.052$, $p = 0.838$). The heatmap in Figure 2 illustrates these relationships across all sub-categories.

In the CG, a similar directional trend was observed, with a weak, non-significant negative correlation between executive queries and final grades ($\rho = -0.260$, $p = 0.281$). As shown in Figure 4, this group's engagement declined after an initial novelty period. While the proportion of executive queries shows a downward trend over time, the pattern is marked by high fluctuation and sparse data in later weeks, preventing any firm conclusions about self-correction. This finding suggests that without guidance, unproductive help-seeking patterns do not reliably resolve on their own.

RQ3: Influence of the Pedagogical Intervention The intervention in the IG served as a clear inflection point, prompting a significant shift in behavior (Figure 3). A Wilcoxon signed-rank test on paired data ($N = 14$) showed that the proportion of executive queries, which had been rising, decreased significantly post-intervention, with a large effect size ($W = 15.0$, $p = 0.036$, $r = 0.560$). This behavioral change was accompanied by a reduction in overall query volume. While the average final exam grade ($M=7.56$, $SD=1.97$) was higher than the midterm grade ($M=7.28$, $SD=2.00$), this improvement was not statistically significant ($W = 34.0$, $p = 0.083$), though the medium-to-large effect size ($r = 0.409$) suggests a potentially meaningful change. This contrasts with the fluctuating, directionless patterns in the CG in Figure 4, reinforcing that the observed shift in the IG was a direct result of the intervention.

4.1 Qualitative Cases of LLM Interaction

Qualitative analysis of the IG's interaction logs provides crucial context for our findings, revealing distinct learner profiles and the nuanced impact of our intervention. High-achieving students demonstrated a variety of effective strategies. Some used the LLM sparingly, relying on strong independent skills, while others effectively used it as a conceptual partner, asking instrumental questions (e.g., "what is the difference between = and ==?") instead of requesting direct fixes.

The intervention's potential is best illustrated by contrasting two students. Mary, who initially relied on executive queries and pasting errors, shifted to asking for general algorithms after the

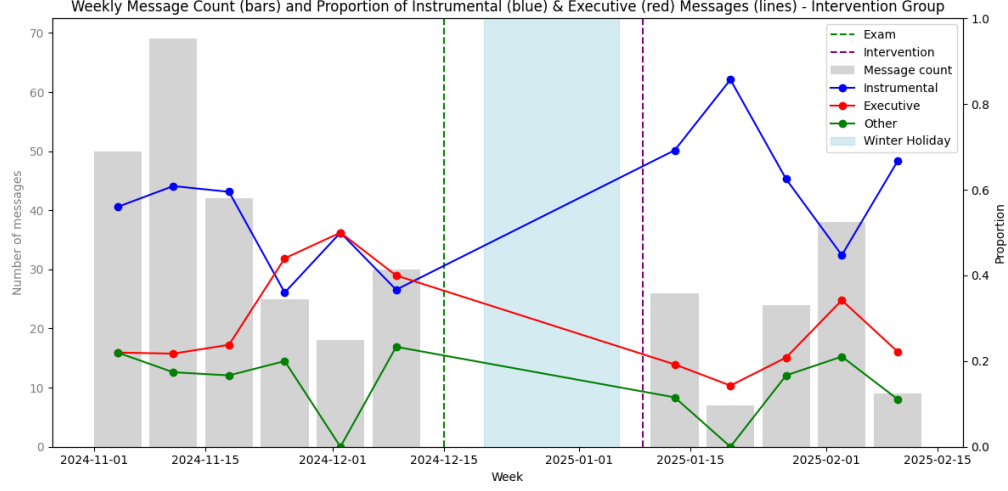


Figure 3: Weekly message counts and proportions per type for the Intervention Group

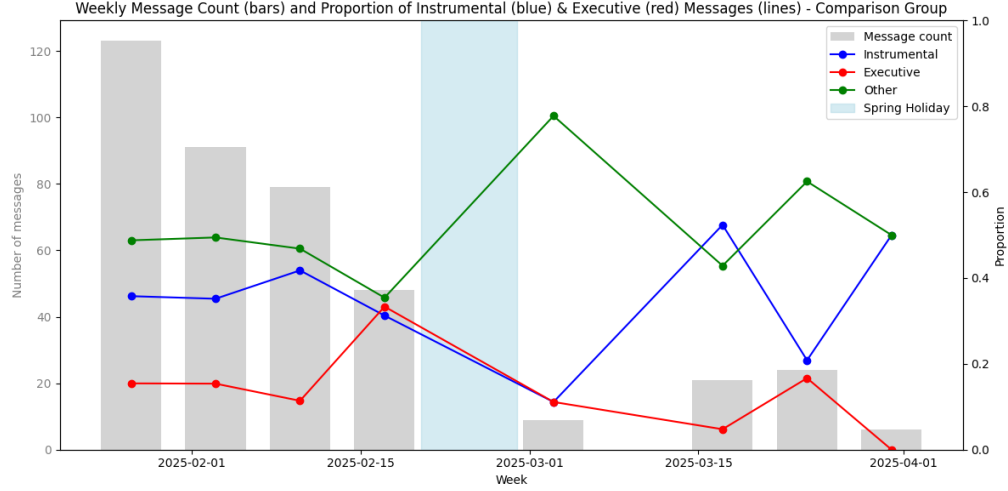


Figure 4: Weekly message counts and proportions per type for the Comparison Group

intervention, leading to improved grades (7.30 $\rightarrow$ 8.65). Mary had a high number of total interactions with Juno and was one of the most active students in terms of code cell executions. Conversely, Sam markedly improved their help-seeking strategy—shifting from 58% executive queries to 64% instrumental—but saw their grades decline (5.50 $\rightarrow$ 4.37). Sam’s case critically suggests that while behavior can be changed, this may not be sufficient to overcome foundational knowledge gaps.

This behavioral shift was mirrored across the group. To understand changes in student workflows, we analyzed sequential patterns of log events using a fixed-window approach, examining the two actions preceding and two actions following each LLM interaction. Pre-intervention, a dominant pattern was reactive and superficial: students would encounter an error, immediately ask the AI for a fix, and then paste the provided code (Edit $\rightarrow$ Error $\rightarrow$ Interaction $\rightarrow$ Edit $\rightarrow$ Success). After the intervention,

this pattern became less frequent. In its place, a more proactive and engaged workflow emerged, where students would successfully execute code and then consult the AI, likely for planning or conceptual clarification, before proceeding with another successful action (Edit $\rightarrow$ Success $\rightarrow$ Interaction $\rightarrow$ Edit $\rightarrow$ Success). This shift from using the AI as a reactive code-fixer to a proactive conceptual partner reinforces the positive behavioral change observed in the quantitative data.

5 Discussion

Addressing RQ1, our finding of a significant negative correlation between unguided LLM usage and exam scores is consistent with prior research [13]. However, our analysis suggests this is not a simple causal relationship, as it contradicts results from similar studies with a more guided LLM programming tool [25, 12]. A

plausible explanation is a selection effect, whereby students with less prior programming experience—who are already more likely to struggle—are the ones self-selecting into higher usage of the tool [2]. Controlling for the self-reported prior experience showed that the initial negative correlation is better explained by this selection effect. This trend is echoed in our qualitative observations; the highest-performing students used the LLM sparingly, whereas students who struggled, such as Sam, demonstrated a higher query volume.

For RQ2, our analysis revealed a sharp contrast: executive help-seeking was detrimental to performance, whereas instrumental help-seeking had a more complex and non-significant correlation. This suggests that sustained high frequency of any help-seeking, even theoretically beneficial instrumental queries, likely indicates persistent difficulty that impacts overall performance [17]. When a struggling student turns to the LLM, a tendency to seek direct solutions rather than explanations will naturally hinder learning and exam performance. This pattern of superficial engagement is supported by our finding that 19.2% of queries in the IG involved copying questions. Such a pattern may then lead to a breakdown in the learning process, where the learner fails to actively process the information provided and just pastes the solution into the notebook [29]. Furthermore, the integration of the LLM may be changing the process from writing code to a more complex co-creation and verification, increasing the challenge for less experienced students [26].

The Comparison Group provides important context, illustrating that these unproductive patterns do not self-correct. Without guidance, the CG's engagement with the chatbot was marked by fluctuating help-seeking proportions and an overall decline in use after an initial novelty period (Figure 4). Instead of converging on productive strategies, students engaged in less-focused conversations, eventually diminishing the perceived usefulness of the tool and reverting to asking their human teacher for assistance—a finding consistent with prior work [24, 16]. This waning engagement likely explains the lack of a clear correlation between their usage and final grades.

Our mid-course intervention for RQ3 successfully guided students away from detrimental executive behaviors and toward more effective instrumental strategies [19]. Post-intervention data indicated success in shifting relative usage: the percentage of executive questions decreased significantly for the group, with a large effect size. And while there was a reduction in use after the novelty of the first 3 weeks, it was not as marked as in the Comparison Group or similar studies [16]. This behavioral shift toward more instrumental approaches was also demonstrated in the post-intervention query patterns of both Mary and Sam (Section 4.1).

5.1 Practical Implications for Educators and System Designers

The findings from this study provide several actionable recommendations for practitioners seeking to effectively integrate LLMs into programming education.

For Educators:

- **Teach Help-Seeking as a Skill:** Do not assume students know how to use LLMs effectively. Our intervention's success suggests that educators should explicitly teach the distinction

between instrumental ("how does this concept work?") and executive ("fix my code") queries.

- **Design for Metacognition:** Assignments should be designed to discourage simple solution-seeking. Instead of tasks that can be solved with a single block of code, educators can create multi-part problems or require students to submit a brief "learning journal" with their code, explaining a key concept they learned from the LLM or a particularly helpful interaction. This incentivizes and makes the learning process visible for students and instructors.
- **Use Analytics for Formative Feedback:** The methods used in this study—analyzing query types and error rates—can serve as a powerful tool for formative assessment. An instructor could review interaction logs to identify groups of students who rely heavily on executive help and provide targeted, individual guidance long before a summative exam.

For System Designers:

- **Prioritize Pedagogy over Task Completion:** Educational chatbots should not be neutral answer-bots. The underlying system prompts must be engineered to be pedagogical agents that prioritize learning. This means adding guardrails to the LLM to prevent it from providing direct solutions and instead guiding the student toward a conceptual understanding, much as a human tutor would.
- **Implement Adaptive Scaffolding:** Systems should be designed to detect patterns of unproductive learning behaviors, like executive help-seeking or help-avoidance. For example, if a student submits consecutive "fix-it" queries, the system could automatically intervene with a metacognitive prompt, such as, "It looks like you're stuck, can you tell me what you were trying to do with this code?" This automates the kind of guidance that proved effective in our intervention.
- **Make Learning Visible to the Learner:** Future iterations of educational programming environments could include a learner-facing dashboard that provides simple analytics (e.g., "This week, most of your questions were productive—great job focusing on improving your understanding!"). This feedback loop empowers students to monitor and regulate their own learning behaviors, fostering the kind of self-awareness demonstrated by our high-achieving participants.

5.2 Contributions

This study makes several distinct contributions. Methodologically, it provides a concrete example of a DBR cycle applied to the integration of generative AI, demonstrating its value in moving from a vaguely defined problem ("students might misuse LLMs") to a specific, evidence-based diagnosis: the negative impact of unguided executive help-seeking. The iterative process of analysis, design, and evaluation refined our understanding, yielding tangible design principles rooted in authentic classroom practice.

Furthermore, the JELAI integration serves as a novel research instrument, capturing granular workflow data (e.g., the sequence of execution after a query) that is lost in studies relying on external chat interfaces. The analytical approach of classifying help-seeking intent (instrumental vs. executive) and analyzing error rates provides a replicable framework for future research.

Empirically, our exploratory study further strengthens evidence that simply providing access to an LLM does not guarantee positive learning outcomes and can, in fact, correlate with lower performance if students adopt non-adaptive strategies. Critically, the qualitative case studies—particularly the divergent outcomes of Mary and Sam—demonstrate that while help-seeking behaviors can be improved through direct intervention, overcoming foundational knowledge gaps presents a significant challenge.

Theoretically, this research extends help-seeking theory into the context of modern generative AI, confirming that the instrumental-executive framework is a powerful lens for understanding student behavior with these tools. By connecting the observed behaviors to established learning science concepts such as the "illusion of understanding" [29], this work highlights the critical need for educational LLMs to be designed not as answer providers, but as pedagogical tools that actively foster student metacognition and critical thinking [9].

5.3 Limitations

This study has several limitations that warrant consideration. First, as an exploratory study, the small sample sizes restrict the generalizability of our findings. The observed negative correlation for specific instrumental comprehension subcategories (as shown in Figure 2) or the lack of significant correlation for total instrumental frequency may be heavily influenced by the specific usage patterns of the lowest- and highest-performing students.

Second, our analysis was confined to interactions with the integrated JELAI chatbot; unmonitored use of external LLMs (like ChatGPT) or other help sources (peers or family) could have confounded the relationship between measured usage and learning outcomes. While students were asked to only use Juno during class, there was no strict enforcement, and they were allowed to chat with each other or ask the teacher for help.

Third, the qualitative analysis of sequential patterns, while indicative, is based on aggregated top frequencies and does not capture workflow changes with rigor. Finally, other individual factors, such as motivation and self-regulation, as well as variations in exam difficulty, may have also influenced the observed trends.

Finally, while this study focused on student help-seeking behaviors, it did not strictly evaluate the accuracy of the LLM's responses. The risk of hallucinations or bias remains a critical challenge in educational AI, one that pedagogical interventions alone cannot resolve, necessitating further technical safeguards [7].

5.4 Future Work

Future work should aim to address these limitations. Replicating this study with larger, more diverse samples is necessary to enhance generalizability and obtain more robust correlation estimates. Research designs could incorporate methods to account for external tool usage and other forms of help-seeking, such as self-reporting or a more detailed log analysis.

Further investigation into sequential patterns could involve more advanced sequence mining techniques on larger datasets to identify macro-trends, or potentially linking specific micro-patterns to learning outcomes at the individual student level [27]. Furthermore, employing experimental designs that control for prior knowledge, motivation, and other individual differences would help isolate

the specific impact of different LLM interaction types and frequencies. Investigating the learning trajectories associated with specific help-seeking patterns, particularly for mid-range performers [11], and exploring optimal LLM design features that effectively scaffold learning remain important avenues for further research [30].

6 Conclusion

This study investigated the role of LLMs in introductory high school programming, finding that they can act as both a learning partner and a hindrance. Our initial results confirm the risks: increased interaction, particularly driven by executive help-seeking, correlated with lower exam performance. While a targeted pedagogical intervention reduced detrimental behaviors, it did not lead to a statistically significant improvement in average grades. Ultimately, this research concludes that the value of an LLM is not inherent to the technology but is critically dependent on pedagogy. Providing access alone is insufficient and can be counterproductive. To ensure LLMs serve as a "friend" and not a "foe," educators and system designers must co-develop learning environments that explicitly teach and scaffold productive, instrumental learning strategies.

Acknowledgments

We thank Thom van der Velden for conducting the intervention and, together with Stijn Risseuw, for assisting with data collection.

References

- [1] Vincent Aleven, Bruce M. McLaren, Jonathan Sewall, Martin van Velsen, Octav Popescu, Sandra Demi, Michael Ringenberg, and Kenneth R. Koedinger. 2016. Example-Tracing Tutors: Intelligent Tutor Development for Non-programmers. *International Journal of Artificial Intelligence in Education*, 26, 1, (Mar. 2016), 224–269. doi:10.1007/s40593-015-0088-2.
- [2] Vincent Aleven, Ido Roll, Bruce M. McLaren, and Kenneth R. Koedinger. 2016. Help Helps, But Only So Much: Research on Help Seeking with Intelligent Tutoring Systems. *International Journal of Artificial Intelligence in Education*, 26, 1, (Mar. 2016), 205–223. doi:10.1007/s40593-015-0089-1.
- [3] Brett A. Becker, Paul Denny, James Finnie-Ansley, Andrew Luxton-Reilly, James Prather, and Eddie Antonio Santos. 2023. Programming Is Hard - Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. ACM, Toronto ON Canada, (Mar. 2023), 500–506. ISBN: 978-1-4503-9431-4. doi:10.1145/3545945.3569759.
- [4] Chin Soon Cheah. 2020. Factors Contributing to the Difficulties in Teaching and Learning of Computer Programming: A Literature Review. *Contemporary Educational Technology*, 12, 2. ERIC Number: EJ1267658. Retrieved Mar. 14, 2025 from <https://eric.ed.gov/?id=EJ1267658>.
- [5] Yung-Ting Chuang and Hsin-Yu Chang. 2024. Analyzing novice and competent programmers' problem-solving behaviors using an automated evaluation system. *Science of Computer Programming*, 237, (Oct. 2024), 103138. doi:10.1016/j.scico.2024.103138.
- [6] Yizhou Fan, Luzhen Tang, Huixiao Le, Kejie Shen, Shufang Tan, Yueying Zhao, Yuan Shen, Xinyu Li, and Dragan Gašević. 2024. Beware of metacognitive laziness: Effects of generative artificial intelligence on learning motivation, processes, and performance. *British Journal of Educational Technology*, n/a, n/a, (Dec. 2024). doi:10.1111/bjet.13544.
- [7] Michail Giannakos, Roger Azevedo, Peter Brusilovsky, Mutlu Cukurova, Yannis Dimitriadis, Davinia Hernandez-Leo, Sanna Järvelä, Manolis Mavrikis, and Bart Rienties. 2024. The promise and challenges of generative AI in education. *Behaviour & Information Technology*, 0, 0, 1–27. doi:10.1080/0144929X.2024.2394886.
- [8] Irene Hou, Sophia Mettillie, Owen Man, Zhuo Li, Cynthia Zastudil, and Stephen MacNeil. 2024. The Effects of Generative AI on Computing Students' Help-Seeking Preferences. In *Proceedings of the 26th Australasian Computing Education Conference (ACE '24)*. Association for Computing Machinery, New York, NY, USA, (Jan. 2024), 39–48. ISBN: 9798400716195. doi:10.1145/3636243.3636248.
- [9] Yu Ji, Zehui Zhan, Tingting Li, Xuanxuan Zou, and Siyuan Lyu. 2025. Human-Machine Cocreation: The Effects of ChatGPT on Students' Learning Performance, AI Awareness, Critical Thinking, and Cognitive Load in a STEM

- Course Toward Entrepreneurship. *IEEE Transactions on Learning Technologies*, 18, 402–415. doi:10.1109/TLT.2025.3554584.
- [10] Stuart A. Karabenick and Myron H. Dembo. 2011. Understanding and facilitating self-regulated help seeking. *New Directions for Teaching and Learning*, 2011, 126, 33–43. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/tl.442>. doi:10.1002/tl.442.
- [11] Shmya Karumbaiah, Jaclyn Ocumpaugh, and Ryan S. Baker. 2022. Context Matters: Differing Implications of Motivation and Help-Seeking in Educational Technology. *International Journal of Artificial Intelligence in Education*, 32, 3, (Sept. 2022), 685–724. doi:10.1007/s40593-021-00272-0.
- [12] 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 2024 CHI Conference on Human Factors in Computing Systems* (CHI '24). Association for Computing Machinery, New York, NY, USA, (May 2024), 1–20. ISBN: 9798400703300. doi:10.1145/3613904.3642773.
- [13] Anjali Khurana, Hariharan Subramonyam, and Parmit K Chilana. 2024. Why and When LLM-Based Assistants Can Go Wrong: Investigating the Effectiveness of Prompt-Based Interactions for Software Help-Seeking. In *Proceedings of the 29th International Conference on Intelligent User Interfaces* (IUI '24). Association for Computing Machinery, New York, NY, USA, (Apr. 2024), 288–303. ISBN: 9798400705083. doi:10.1145/3640543.3645200.
- [14] Shao-Heng Ko and Kristin Stephens-Martinez. 2024. The Trees in the Forest: Characterizing Computing Students' Individual Help-Seeking Approaches. In *Proceedings of the 2024 ACM Conference on International Computing Education Research - Volume 1* (ICER '24). Vol. 1. Association for Computing Machinery, New York, NY, USA, (Aug. 2024), 343–358. ISBN: 9798400704758. doi:10.1145/3632620.3671099.
- [15] Anastassis Kozanitis, Jean-Francois Desbiens, and Roch Chouinard. 2007. Perception of Teacher Support and Reaction towards Questioning: Its Relation to Instrumental Help-Seeking and Motivation to Learn. *International Journal of Teaching and Learning in Higher Education*, 19, 3, 238–250. Publisher: International Society for Exploring Teaching and Learning ERIC Number: EJ901297. Retrieved Oct. 7, 2025 from <https://eric.ed.gov/?id=EJ901297>.
- [16] Wenhan Lyu, Yimeng Wang, Tingting (Rachel) Chung, Yifan Sun, and Yixuan Zhang. 2024. Evaluating the Effectiveness of LLMs in Introductory Computer Science Education: A Semester-Long Field Study. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale* (L@S '24). Association for Computing Machinery, New York, NY, USA, (July 2024), 63–74. ISBN: 979-8-4007-0633-2. doi:10.1145/3657604.3662036.
- [17] Samiha Marwan, Anay Dombé, and Thomas W. Price. 2020. Unproductive Help-seeking in Programming: What it is and How to Address it. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (ITICSE '20). Association for Computing Machinery, New York, NY, USA, (June 2020), 54–60. ISBN: 978-1-4503-6874-2. doi:10.1145/3341525.3387394.
- [18] Sharon Nelson-Le Gall. 1981. Help-seeking: An understudied problem-solving skill in children. *Developmental Review*, 1, 3, (Sept. 1981), 224–246. doi:10.1016/0273-2297(81)90019-8.
- [19] Davy Tsz Kit Ng, Chee Wei Tan, and Jac Ka Lok Leung. 2024. Empowering student self-regulated learning and science education through ChatGPT: A pioneering pilot study. *British Journal of Educational Technology*, 55, 4, 1328–1353. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/bjet.13454>. doi:10.1111/bjet.13454.
- [20] Sankalan Pal Chowdhury, Vilém Zouhar, and Mrinmaya Sachan. 2024. AutoTutor meets Large Language Models: A Language Model Tutor with Rich Pedagogy and Guardrails. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale* (L@S '24). Association for Computing Machinery, New York, NY, USA, (July 2024), 5–15. ISBN: 9798400706332. doi:10.1145/3657604.3662041.
- [21] Minju Park, Sojung Kim, Seunghyun Lee, Soonwoo Kwon, and Kyuseok Kim. 2024. Empowering Personalized Learning through a Conversation-based Tutoring System with Student Modeling. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (CHI EA '24). Association for Computing Machinery, New York, NY, USA, (May 2024), 1–10. ISBN: 979-8-4007-0331-7. doi:10.1145/3613905.3651122.
- [22] Thomas W. Price, Zhongxiu Liu, Veronica Cateté, and Tiffany Barnes. 2017. Factors Influencing Students' Help-Seeking Behavior while Programming with Human and Computer Tutors. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (ICER '17). Association for Computing Machinery, New York, NY, USA, (Aug. 2017), 127–135. ISBN: 978-1-4503-4968-0. doi:10.1145/3105726.3106179.
- [23] Ido Roll, Ryan S. J. d. Baker, Vincent Alevan, and Kenneth R. Koedinger. 2014. On the Benefits of Seeking (and Avoiding) Help in Online Problem-Solving Environments. *Journal of the Learning Sciences*, 23, 4, (Oct. 2014), 537–560. Publisher: Routledge eprint: <https://doi.org/10.1080/10508406.2014.883977>. doi:10.1080/10508406.2014.883977.
- [24] Mika Setälä, Ville Heilala, Pieta Sikström, and Tommi Kärrkäinen. 2025. The Use of Generative Artificial Intelligence for Upper Secondary Mathematics Education Through the Lens of Technology Acceptance. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing* (SAC '25). Association for Computing Machinery, New York, NY, USA, (May 2025), 74–82. ISBN: 979-8-4007-0629-5. doi:10.1145/3672608.3707817.
- [25] Brad Sheese, Mark Liffiton, Jaromir Savelka, and Paul Denny. 2024. Patterns of Student Help-Seeking When Using a Large Language Model-Powered Programming Assistant. In *Proceedings of the 26th Australasian Computing Education Conference* (ACE '24). Association for Computing Machinery, New York, NY, USA, (Jan. 2024), 49–57. ISBN: 9798400716195. doi:10.1145/3636243.3636249.
- [26] Chad C. Tossell, Nathan L. Tenhundfeld, Ali Momen, Katrina Cooley, and Ewart J. de Visser. 2024. Student Perceptions of ChatGPT Use in a College Essay Assignment: Implications for Learning, Grading, and Trust in Artificial Intelligence. *IEEE Transactions on Learning Technologies*, 17, 1069–1081. doi:10.1109/TLT.2024.3355015.
- [27] Manuel Valle Torre, Catharine Oertel, and Marcus Specht. 2024. The Sequence Matters in Learning - A Systematic Literature Review. In *Proceedings of the 14th Learning Analytics and Knowledge Conference* (LAK '24). Association for Computing Machinery, New York, NY, USA, (Mar. 2024), 263–272. ISBN: 9798400716188. doi:10.1145/3636555.3636880.
- [28] Manuel Valle Torre, Thom van der Velden, Marcus Specht, and Catharine Oertel. 2025. JELAI: Integrating AI and Learning Analytics in Jupyter Notebooks. In *Artificial Intelligence in Education*. Alexandra I. Cristea, Erin Walker, Yu Lu, Olga C. Santos, and Seiji Isotani, (Eds.), Springer Nature Switzerland, Cham, 68–75. ISBN: 978-3-031-98465-5. doi:10.1007/978-3-031-98465-5_9.
- [29] Jörg Wittwer and Alexander Renkl. 2008. Why Instructional Explanations Often Do Not Work: A Framework for Understanding the Effectiveness of Instructional Explanations. *Educational Psychologist*, 43, 1, (Jan. 2008), 49–64. Publisher: Routledge eprint: <https://doi.org/10.1080/00461520701756420>. doi:10.1080/00461520701756420.
- [30] Rong Wu and Zhonggen Yu. 2024. Do AI chatbots improve students learning outcomes? Evidence from a meta-analysis. *British Journal of Educational Technology*, 55, 1, 10–33. <https://onlinelibrary.wiley.com/doi/pdf/10.1111/bjet.13334>. doi:10.1111/bjet.13334.
- [31] Ruiwei Xiao, Xinying Hou, Harsh Kumar, Steven Moore, John Stamper, and Michael Liut. 2024. A Preliminary Analysis of Students' Help Requests with an LLM-powered Chatbot when Completing CS1 Assignments. In *8th Educational Data Mining in Computer Science Education Workshop*. Atlanta, GA, (July 2024).
- [32] Benjamin Xie, Jared Ordonia Lim, Paul K.D. Pham, Min Li, and Amy J. Ko. 2023. Developing Novice Programmers' Self-Regulation Skills with Code Replays. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1* (ICER '23). Vol. 1. Association for Computing Machinery, New York, NY, USA, (Sept. 2023), 298–313. ISBN: 978-1-4503-9976-0. doi:10.1145/3568813.3600127.
- [33] Fan Yang and Jill Stefaniak. 2023. A Systematic Review of Studies Exploring Help-Seeking Strategies in Online Learning Environments. *Online Learning*, 27, 1, (Mar. 2023). doi:10.24059/olj.v27i1.3400.
- [34] Sine Zambach. 2025. AI-Enhanced Learning: Comparing Outcomes in Introductory and Advanced Programming Courses. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing* (SAC '25). Association for Computing Machinery, New York, NY, USA, (May 2025), 104–105. ISBN: 979-8-4007-0629-5. doi:10.1145/3672608.3707909.
- [35] Chunpeng Zhai, Santoso Wibowo, and Lily D. Li. 2024. The effects of over-reliance on AI dialogue systems on students' cognitive abilities: a systematic review. *Smart Learning Environments*, 11, 1, (June 2024), 28. doi:10.1186/s40561-024-00316-7.
- [36] Xin Zhang, Peng Zhang, Yuan Shen, Min Liu, Qiong Wang, Dragan Gašević, and Yizhou Fan. 2024. A Systematic Literature Review of Empirical Research on Applying Generative Artificial Intelligence in Education. *Frontiers of Digital Education*, 1, 3, (Sept. 2024), 223–245. doi:10.1007/s44366-024-0028-5.